UNIVERSIDAD AUTÓNOMA METROPOLITANA

Casa abierta al tiempo PLANTEL IZTAPALAPA

PROYECTO TERMINAL

GRABADOR DE EPROMs AUTÓNOMO

REALIZADO POR:

JOSÉ IGNACIO FLORES TAPIA.
NÉSTOR SÁNCHEZ GÓMEZ.

CON ASESORÍA DE:

MIGUEL ÁNGEL RUÍZ SÁNCHEZ

PROLOGO

El proyecto presente se dividió para su análisis y diseño en tres bloques principales, que son:

1. *HARDWARE*
2. *SOFTWARE*
3. *MANUAL DE USUARIO*

Para el bloque de HARDWARE tenemos:

Análisis: En esta etapa se analizan los requerimientos del sistema grabador, y en base a estos se proponen varias soluciones y alternativas.
Diseño: Aquí se realiza un bosquejo en el cual se proporciona una descripción de los diferentes circuitos utilizados de acuerdo a los requerimientos de la etapa anterior. Se parte de la configuración de los pines de las EPROMs, después se analizan los circuitos de control para cada uno de los pines, a continuación se describen los componentes para la interfaz con los anteriores, y por último se analiza el microcontrolador usado y los periféricos que son memoria externa RAM y ROM, así como los latch, buffers, y decodificadores para el control de los periféricos.
Implementación: En esta parte se proporciona una lista de los principales circuitos integrados que se requerirán para el grabador con una descripción y una justificación de porque se usaron. Además del diagrama final simbólico y un diagrama con la distribución física.

Para el bloque de SOFTWARE tenemos:

Análisis: Para llevar acabo esta etapa se toma en cuenta el bloque anterior, en el cual se analizan las necesidades y las posibles formas de cubrirlas.
Diseño: Se proyectan soluciones (en diagramas de flujo) en base a las diferentes funciones que será capaz de desarrollar el sistema grabador.
Implementación: Aquí se proporciona una relación de los códigos fuentes para dar solución a cada una de las funciones del grabador.

En el bloque del MANUAL DE USUARIO se proporciona una guía paso a paso de cómo debe ser operado el grabador por el usuario.

CONTENIDO

Justificación.

En el presente proyecto se realizará el diseño de un sistema grabador de EPROMs autónomo de una PC. Esto es que pueda grabar de EPROM a EPROM sin que se tenga que hacer uso de una PC, además de que también se podrá usar para grabar de una PC a una EPROM a través de un RS232 que es una interfaz por el puerto serial.

En un grabador convencional se pueden realizar varias operaciones, entre las mas elementales están:

CARGAR: El programador acepta un archivo en código HEX, y lo carga en su buffer de memoria.
DESPLEGAR: Se despliega el contenido del buffer de memoria del programador en la pantalla de la PC.
COPIAR: Se lee el contenido de la EPROM y se almacena en el buffer de memoria del programador.
PROGRAMAR: Se almacena el contenido del buffer de memoria del programador en el arreglo de memoria de la EPROM.
COMPARAR: Se compara el contenido de la EPROM con el contenido del buffer de memoria del programador.
VERIFICAR: Se examina el contenido de la EPROM y se determina si esta ha sido borrada.
TIPO: Se selecciona el tipo de EPROM que será insertada en el socket del programador.
EDITAR: Se puede alterar el contenido del buffer de memoria del programador.
GRABAR: Comprende los procesos de COPIAR y PROGRAMAR,

Cada que se requiere realizar alguna de estas operaciones, por fuerza es necesario hacer uso de una computadora, en la cual este instalado el software y el hardware del grabador, pero ¿que pasa si necesitamos únicamente realizar varias copias del contenido de una misma EPROM?, o si queremos comparar el contenido de una EPROM con el contenido de la otra. En estos casos usar una PC únicamente para este propósito es un desperdicio, podemos entonces pensar en el diseño de un grabador de EPROMs que sea autónomo de una PC y que nos sirva para estos propósitos, con la ventaja de que en cualquier momento que se requiera realizar alguna operación adicional a estas podemos conectar el grabador a una PC cualquiera y vía puerto serial llevar acabo nuestros propósitos.

Objetivo.

Diseñar e implementar un grabador de EPROMs autónomo (Que copie de EPROM a EPROM, además de PC a EPROM).

Características:

El grabador tiene capacidad para grabar EPROMs de hasta 64Kb, que es mucho mas de las necesidades cotidianas en un laboratorio de Sistemas Digitales, esto es, grabará EPROMs de la forma 27XXX, que va desde la 2716 (2Kb) hasta la 27512 (64Kb).

Se pueden hacer copias múltiples de una sola EPROM, es decir, el contenido de la EPROM fuente se vacía en la RAM del sistema y después el contenido de la RAM se puede vaciar el contenido en las N EPROMs destino que se deseen.

Para realizar lo anterior se contará con un solo socket, y el programa irá pidiendo al usuario la EPROM fuente y la EPROM destino cuando sean requeridas

El sistema grabador funcionara tanto en forma autónoma (para realizar copias), como es forma dependiente (interfazarse con un PC a través de un puerto serial).

INTRODUCCIÓN.

En los sistemas digitales capaces de trabajar independientemente (tales como una PC), existe un lugar donde se encuentran localizadas las instrucciones que permitirán que dicho sistema realice las funciones para las cuales fue diseñado, dicho lugar es una memoria.

Debido a que seria muy costoso mantener el sistema siempre funcionando con su programa en una memoria RAM (Random Access Memories), ya que estas pierden su información al dejar de existir un voltaje de alimentación surgen memorias que permiten mantener información aun sin que exista alguna señal de alimentación, dichas memorias son llamadas EPROM (Erasable Programmable Read Only Memories).

Estas memorias, como ya se comento, mantienen datos en ellas aun sin voltaje de alimentación presente permitiendo apagar el sistema cuando no se esté utilizando; en contraste con lo anterior, la forma en la cual se les introduce la información es diferente que en una RAM convencional. Para grabar información en una memoria EPROM se hace necesario un sistema especial que nos ayude a realizar esta tarea llamado GRABADOR o también PROGRAMADOR.

Para grabar las memorias EPROM se hace necesario mandarle no solo direcciones y datos, sino también otras señales de control, tales como Chip Enable (CE), Output Enable (OE) y Programming Pulse Voltage (Vpp) siendo este último un voltaje alto (normalmente 25 V o 21 V o 12.5 V según el tipo de EPROM) comparado con lógica TTL (0 V y 5 V).

Los grabadores de EPROM mas conocidos, son aquellos que dependen de una computadora (funciones del microprocesador, memoria RAM, etc), pero en el presente proyecto se intentara el diseño de un grabador autónomo.

ANÁLISIS Y DISEÑO DEL HARDWARE.

Dado que lo que se necesita es diseñar un grabador de EPROMs, lo primero que se debe de analizar es la configuración de los pines para cada una de estas memorias. En base a un análisis de las terminales que presentan los diferentes modelos de EPROMs, con los cuales se podrá trabajar, que van desde la 2716 hasta la 27512, se observó que existen 7 terminales que difieren, tal como se observa en la siguiente tabla:

27512	27256	27128	2764	2732	2716	# PIN	#PIN	2716	2732	2764	27128	27256	27512
A15	Vpp	Vpp	Vpp			1	28			Vcc	Vcc	Vcc	Vcc
A12	A12	A12	A12			2	27			PGM	PGM	A14	A14
A7	A7	A7	A7	A7	A7	3	26	Vcc	Vcc	N.C.	A13	A13	A13
A6	A6	A6	A6	A6	A6	4	25	A8	A8	A8	A8	A8	A8
A5	A5	A5	A5	A5	A5	5	24	A9	A9	A9	A9	A9	A9
A4	A4	A4	A4	A4	A4	6	23	Vpp	A11	A11	A11	A11	A11
A3	A3	A3	A3	A3	A3	7	22	OE	OE/Vpp	OE	OE	OE	OE/Vpp
A2	A2	A2	A2	A2	A2	8	21	A10	A10	A10	A10	A10	A10
A1	A1	A1	A1	A1	A1	9	20	CE	CE	CE	CE	CE	CE
A0	A0	A0	A0	A0	A0	10	19	O7	O7	O7	O7	O7	O7
O0	O0	O0	O0	O0	O0	11	18	O6	O6	O6	O6	O6	O6
O1	O1	O1	O1	O1	O1	12	17	O5	O5	O5	O5	O5	O5
O2	O2	O2	O2	O2	O2	13	16	O4	O4	O4	O4	O4	O4
GND	GND	GND	GND	GND	GND	14	15	O3	O3	O3	O3	O3	O3

Como puede observarse, algunos modelos son de 28 pines y otros de 24. Analizando la figura se obtiene la siguiente relación de voltajes en la que podemos observar de acuerdo al tipo de EPROM que voltaje necesita para cada uno de los pines en los cuales se aplica algún voltaje.

#PIN	2716	2732	2764	27128	27256	27512
1	N/C	N/C	Vpp (21/12.5/5)	Vpp (21/12.5/5)	Vpp (12.5/5)	A15 (5/0)
28	N/C	N/C	Vcc (6/5)	Vcc (6/5)	Vcc (6/5)	Vcc (6/5)
27	N/C	N/C	PGM (5/0)	PGM (5/0)	A14 (5/0)	A14 (5/0)
26/24	Vcc (5)	Vcc (5)	N/C	A13 (5/0)	A13 (5/0)	A13 (5/0)
23 /21	Vpp (25/5)	A11 (5/0)	A11 (5/0)	A11 (5/0)	A11 (5/0)	A11 (5/0)
22/20	OE (5/0)	OE/Vpp (25/21/0)	OE (5/0)	OE (5/0)	OE (5/0)	OE/Vpp (5/0)
20/18	CE (5/0)	CE (5/0)	CE (5/0)	CE (5/0)	CE (5/0)	CE (5/0)

NOTA: En la columna del #PIN se observan expresiones para dos números diferentes de pines, esto es porque los pines son diferentes dependiendo si son EPROMs de 24 o de 28 pines. Los números entre paréntesis de las demás columnas se refieren a los diferentes voltajes que se necesitan en esos pines.

De la relación anterior se observa que para los pines 20 y 27, no existe un problema complicado para las diferentes señales según el tipo de EPROM, en comparación con los pines restantes. Para estos últimos fue necesario diseñar un circuito que pudiera

ser controlado directamente por el programa, para proporcionar la señal necesaria, por lo cual se desarrollo el circuito de la figura 1 (en base a el circuito de la figura 2 en el cual se observan las consideraciones de corriente y de voltaje), que utilizan reguladores de voltaje LM317 entre otros elementos.

FIGURA 1. CIRCUITO PARA CONTROLAR LOS PINES DEL SOCKET.

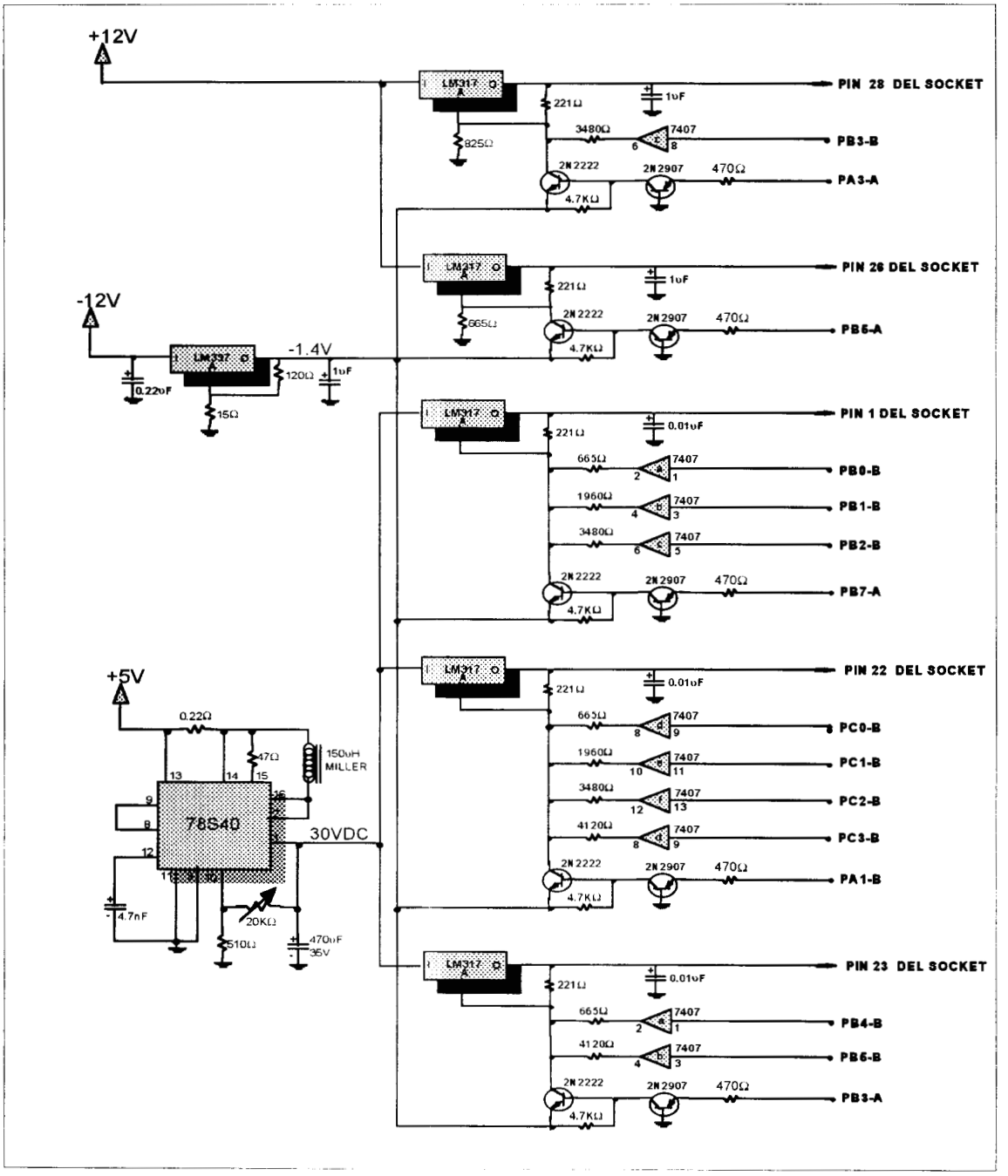
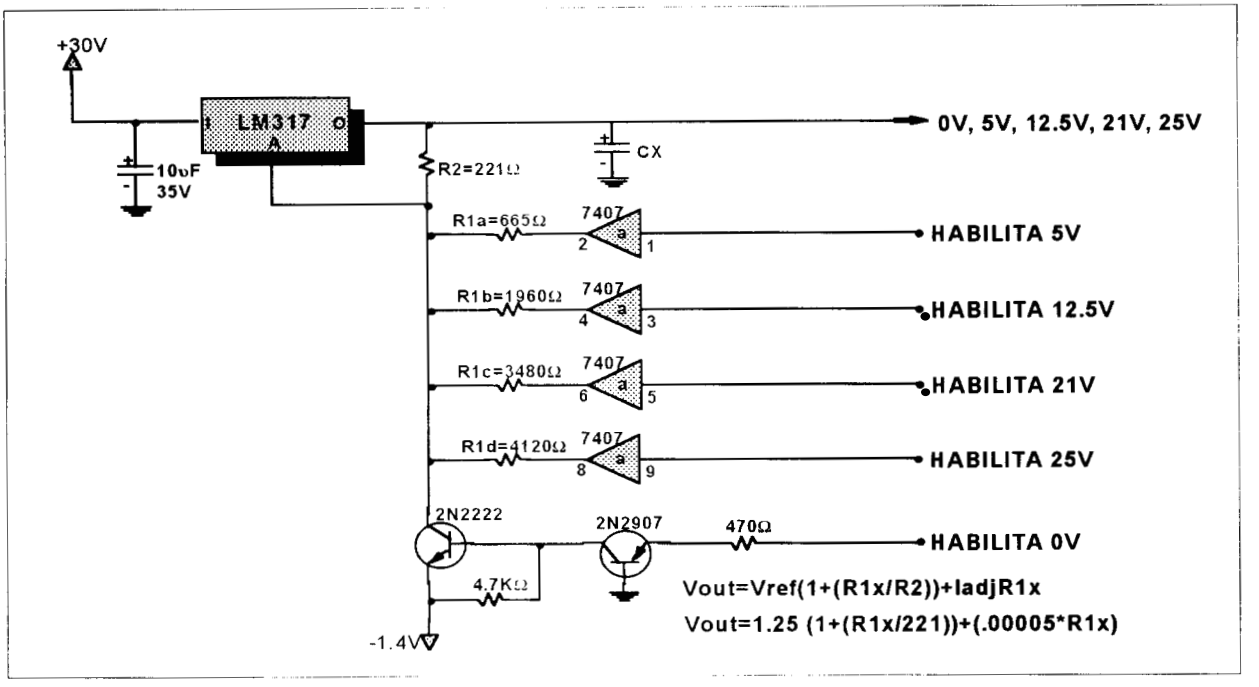
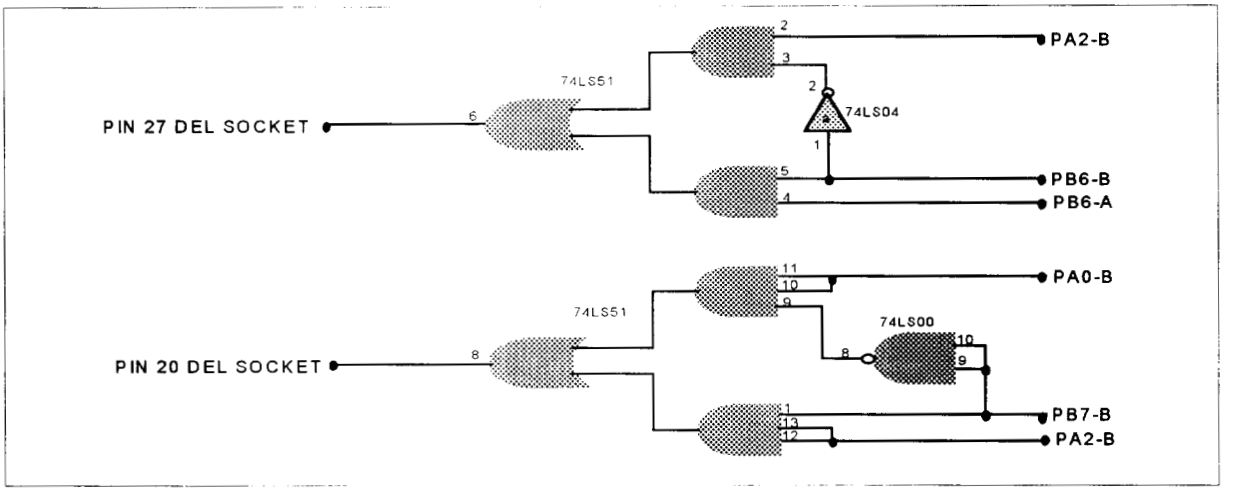


FIGURA 2. CIRCUITO BASE PARA CONTROLAR PINES DEL SOCKET



Los pines 20 y 27 utilizan un circuito combinacional mas sencillo que puede observarse en el circuito de la figura 3.

FIGURA 3. CIRCUITO COMBINACIONAL PARA CONTROL DE PINES 20 Y 27.



De esta forma para activar algún voltaje del circuito de control de voltajes, basta mandar un cero lógico y para activar cero volts se precisa de un uno lógico, lo cual hace posible su control desde el programa principal.

Para llevar acabo dicho control se precisa de un circuito que organice todo el funcionamiento, es aquí donde se hace necesario el uso de un microcontrolador, entre los circuitos considerados están: 8031, 8051, 8032, 8052, todos ellos de Intel. Una de la características para decidir entre ellos fueron los requerimientos de memoria ROM

interna, externa o ambas que se tienen en el proyecto, además del costo y la existencia de ellos en el mercado..

Como la programación se realizará en lenguaje ensamblador, no se necesita de ningún interprete dentro del micro, además de que algunas funciones extras que se presentan en los micros 8051, 8032 y 8052, como son las interrupciones, no serán usados en el presente proyecto.

Tomando en cuenta una breve investigación de mercado, se obtuvo como resultado que el microcontrolador 8031 es mas barato que el 8051 casi en una tercera parte, y que el 8052 es un micro que esta discontinuado, por lo cual se resuelve utilizar el microcontrolador 8031 para el grabador.

El microcontrolador se encargará de proporcionar las "señales" para leer la EPROM fuente, grabar la EPROM destino, etc. Para llevar acabo esta entrega de "señales" de precisa de algunos pasos intermedios, como lo es almacenar los datos fuentes en la memoria RAM del sistema, además de realizar el direccionamiento de los datos (poner los datos en las localidades de memoria correctas).

Entre los posibles circuitos auxiliares a utilizar , se encuentran algunos buffers, unas PPIs (Programmable Periferical Interfaz), las cuales serán las encargadas de proporcionar los mensajes adecuados al socket, y para la interfaz con el usuario se precisará de circuitería, como son decodificadores, displays de 7 segmentos, y circuitería combinacional y secuencial para control y manipulación del grabador. Un diagrama inicial de lo que será el grabador de EPROMs se observa en la figura 4.

Para el grabador se tiene proyectado usar una EPROM que sea la encargada de almacenar el software propio del grabador, debido a esto no se hace necesario que el microcontrolador a usar posea memoria ROM interna.

Para la parte de la interfaz con el usuario se propone el uso de un display de 7 segmentos (será el que desplegara mayor información del estado de los procesos que se realicen en el grabador), un par de leds que indicaran cuando se puede remover una EPROM del socket y cuando no, además de un tercero que indicara si se esta trabajando en modo dependiente o en modo dependiente, y un par de botones. El primer botón (PB2) se usará para navegar en todos los menús y submenús. El segundo botón (PB1) será para iniciar el proceso seleccionado por PB2.

El circuito propuesto para esta parte se muestra en la figura 5.

Cabe señalar que para la interfaz de usuario se podría haber usado un número mayor de displays y mas botones, o inclusive un LCD y un teclado controlados por un 8279, en este proyecto no se realizo de esa manera, pero queda abierto para posteriores modificaciones, además de que podría servir muy bien para efectos de un sistema de desarrollo del microcontrolador 8031 realizando las modificaciones pertinentes.

Con lo que se ha descrito hasta este momento el grabador de EPROMs funcionará de forma autónoma, pero si se quiere usar de forma dependiente, es decir, como un grabador convencional, debemos de considerar algunos aspectos extras, como son:

¿Cómo se implementará la comunicación entre el grabador y la PC?,

¿Cómo debe de ser el programa que controle al grabador? y

¿Cómo será el programa que controle a la PC?.

Para responder a estas preguntas de debe de analizar, con que es con lo que se cuenta en cuanto a hardware y en cuanto a software, y ver si con esto es suficiente o si necesitamos implementar algo mas.

Evidentemente para solucionar el problema de comunicación entre la PC y el grabador, dadas las características del microcontrolador que cuenta con las salidas adecuadas para este fin (TXD y RXD), se puede usar un RS232, que es una interfaz serial,. Lo que se debe de hacer entonces es implementar el hardware necesario, en la figura 6 se muestran las conexiones para este propósito.

FIGURA 4. DIAGRAMA INICIAL DEL GRABADOR

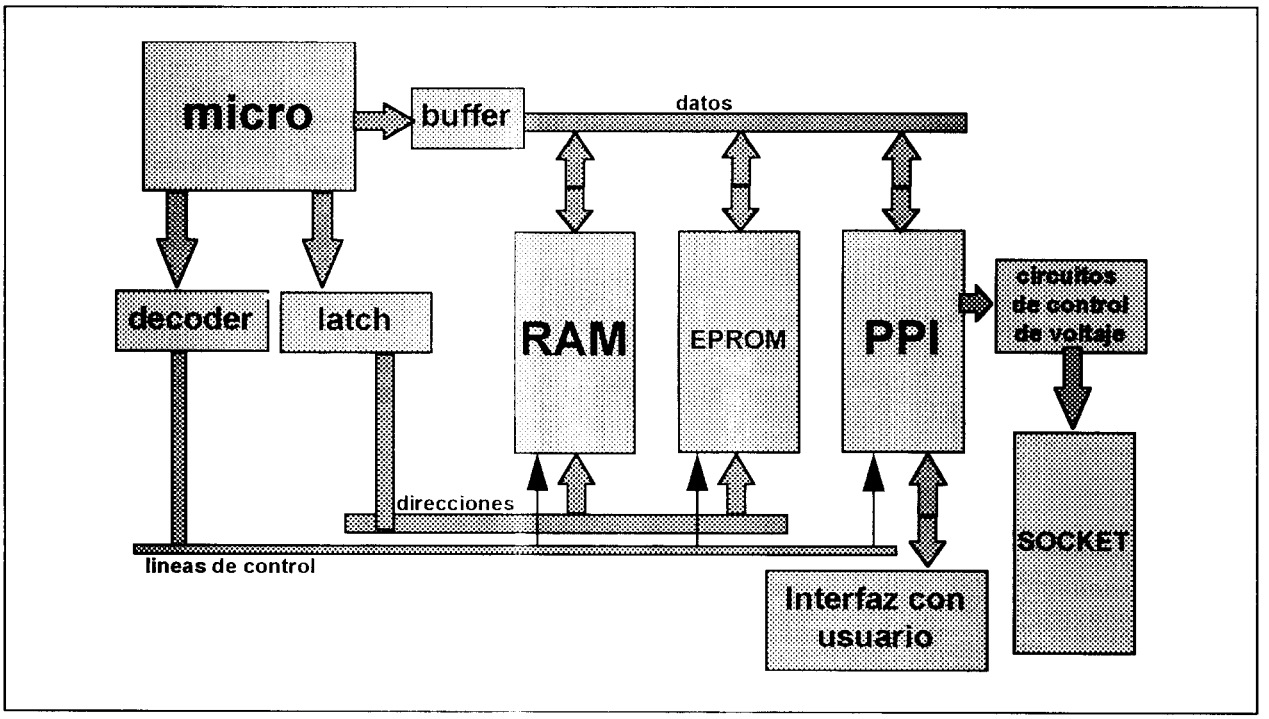


FIGURA 5. CIRCUITO PROPUESTO PARA INTERFAZ CON EL USUARIO

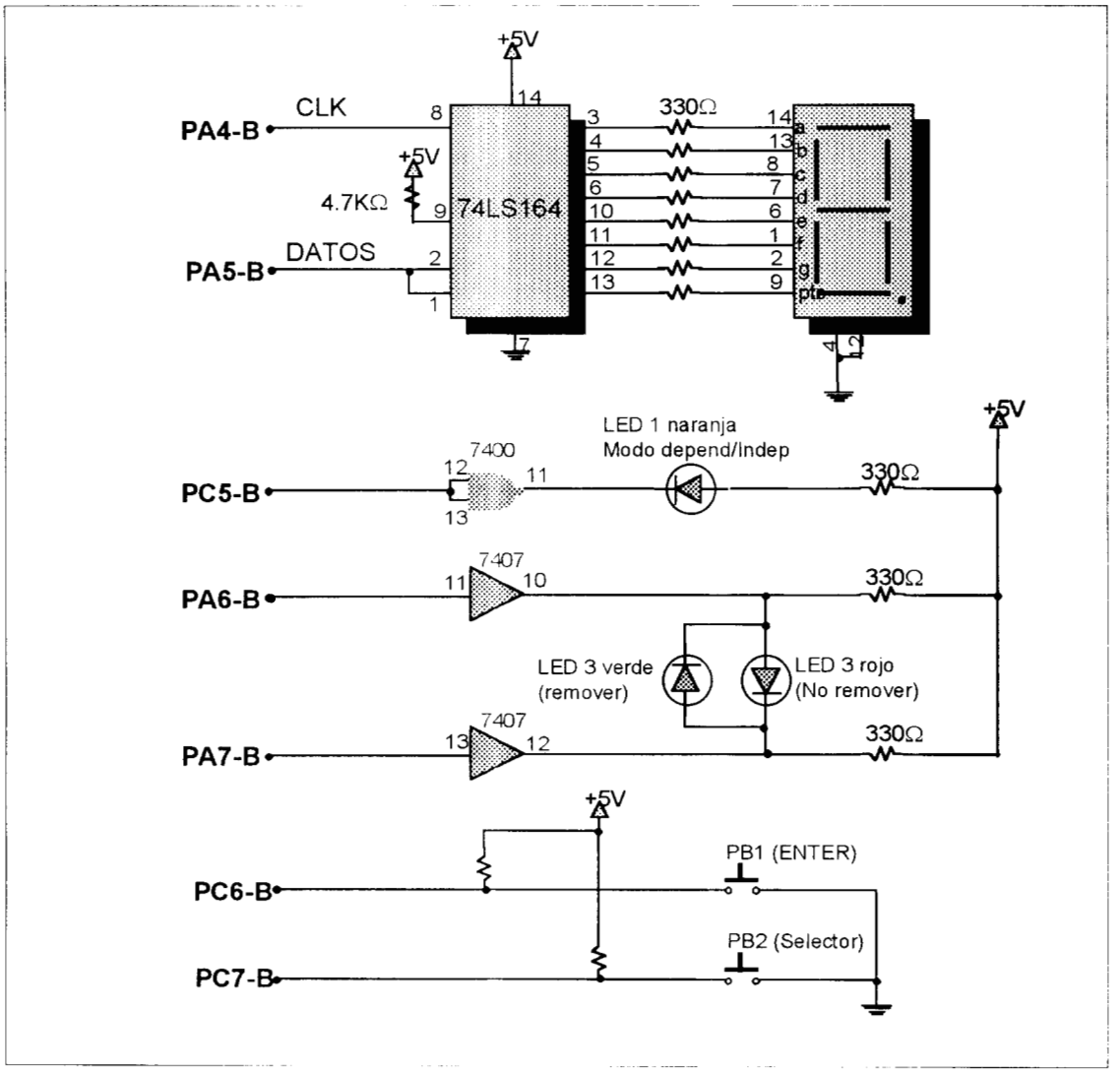
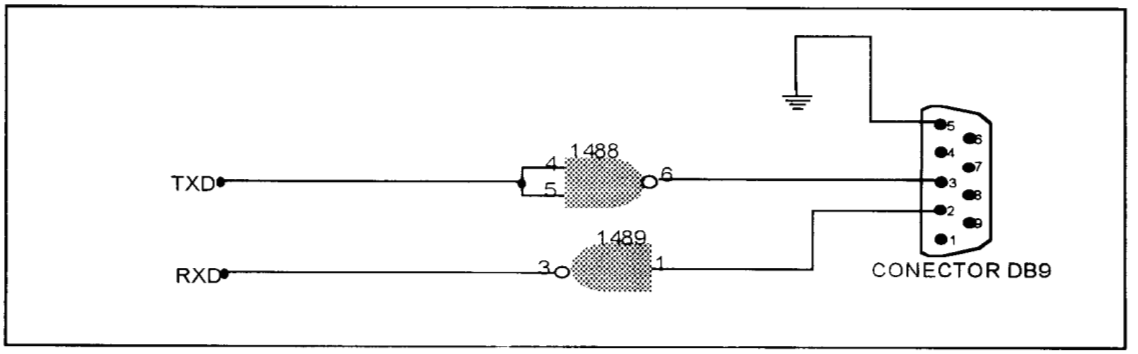


FIGURA 6. Interfaz serial RS232

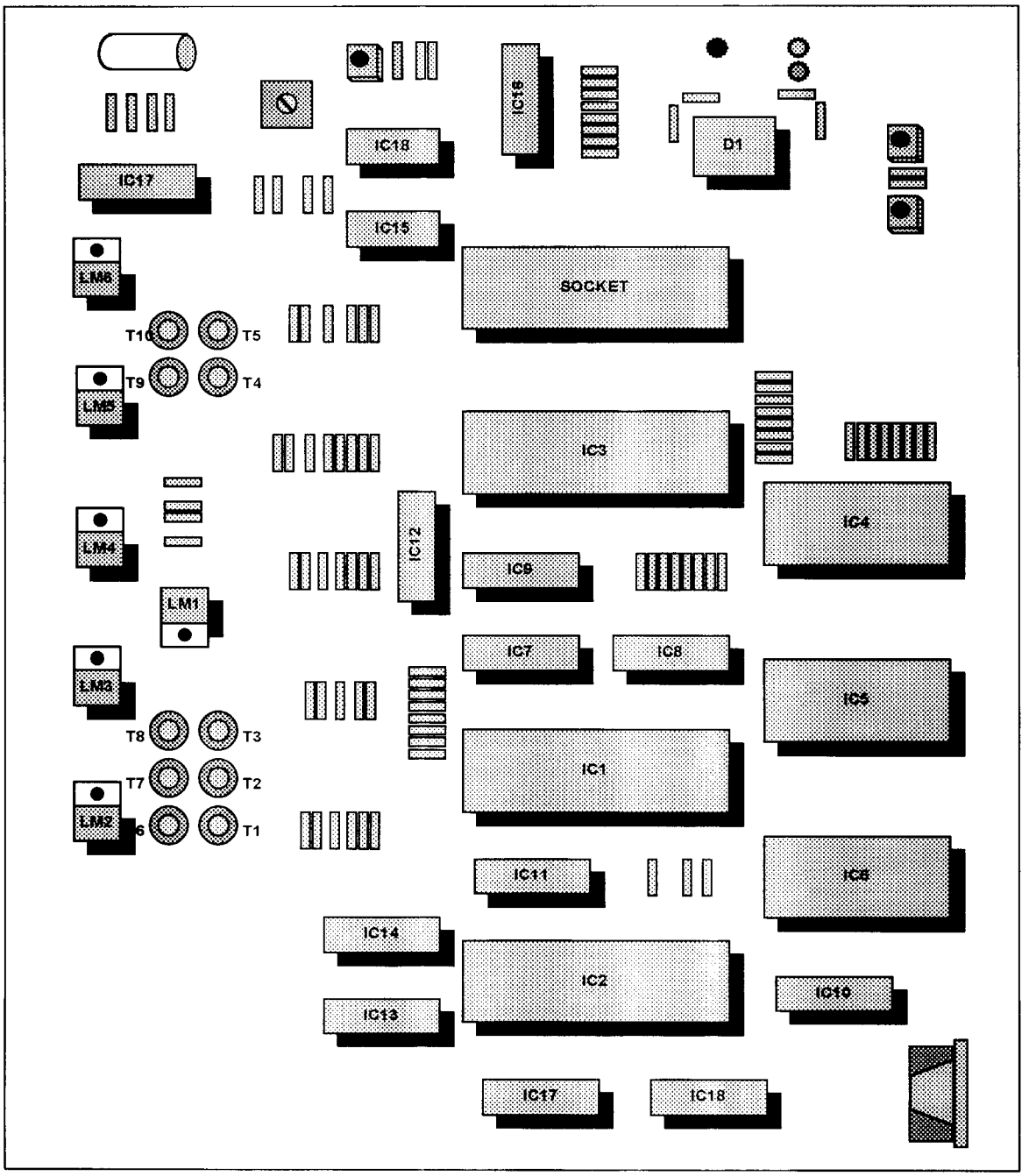


IMPLEMENTACIÓN DEL HARDWARE.

Después de todo este análisis, ya se puede decidir que componentes se utilizarán para llevar acabo el proyecto. En la figura 7 se presenta la distribución física de estos en el grabador de EPROMs, en la figura 8 hay una lista de todos los circuitos y componentes usados y en el ANEXO 1 se proporciona una breve justificación para cada uno, además de las hojas técnicas de los C.I. (Circuitos Integrados) mas importantes.

A continuación se presentan los diagramas correspondientes a los bloques de la figura 4, que son los circuitos con los cuales se implemento el grabador.

FIGURA 7 DISTRIBUCIÓN FÍSICA DE LOS COMPONENTES.



LISTA DE LOS COMPONENTES USADOS

IC- NÚMERO	NÚMERO ID	DESCRIPCIÓN
ICE	8031	Microcontrolador de 8 bits
IC2	8255-A	Interfaz periférica programable
IC3	8255-B	Interfaz periférica programable
IC4	62256	RAM de 32kb x 8bits
IC5	6264	RAM de 8Kb x 8bits
IC6	2764	EPROM de 8Kb x 8bits
IC7	74LS373	Latch 8 bits
IC8	74LS245-A	Buffer bidireccional de 8 bits
IC9	74LS245-B	Buffer bidireccional de 8 bits
IC10	74LS138	Decodificador de 3 a 8
IC11	74LS04	Compuertas lógicas NOT
IC12	74LS08	Compuertas lógicas AND
IC13	7407	Buffers para DC mayor a TTL
IC14	7407	Buffers para DC mayor a TTL
IC15	74LS00	Compuertas lógicas NAND
IC16	MUA78S40	Convertidor de DC a DC
IC17	1488	Convertidor de TTL a RS232
IC18	1489	Convertidor de RS232 a TTL
IC19	74LS51	Compuerta AND-OR-NEGADA con dos compuertas de dos y tres entradas.
IC20	74LS164	Registro de entrada serie/salida paralela
D1	MAN7409	Display de 7 segmentos
LM1	LM337	Regulador de Voltaje
LM2-LM6	LM317	Regulador de Voltaje
T1-T5	2N2222	Transistores
T6-T10	2N2902	Transistores
R4.7	4.7 K	Resistencias de 4.7K para efectos de PULL-OP
RX:X	Varios	Resistencias varias
CAPX.X	Varios	Capacitores varios
POT1	20K	Potenciómetro de precisión de 20K
LEDX	led	Leds
TMR1	11.0592mhz	Cristal para velocidad del KIT
BOB1	150 microH	Bobina de Miller

FIGURA 9. BLOQUE PARA EL MICROCONTROLADOR, LOS BUFFERS, EL LATCH Y EL DECODIFICADOR.

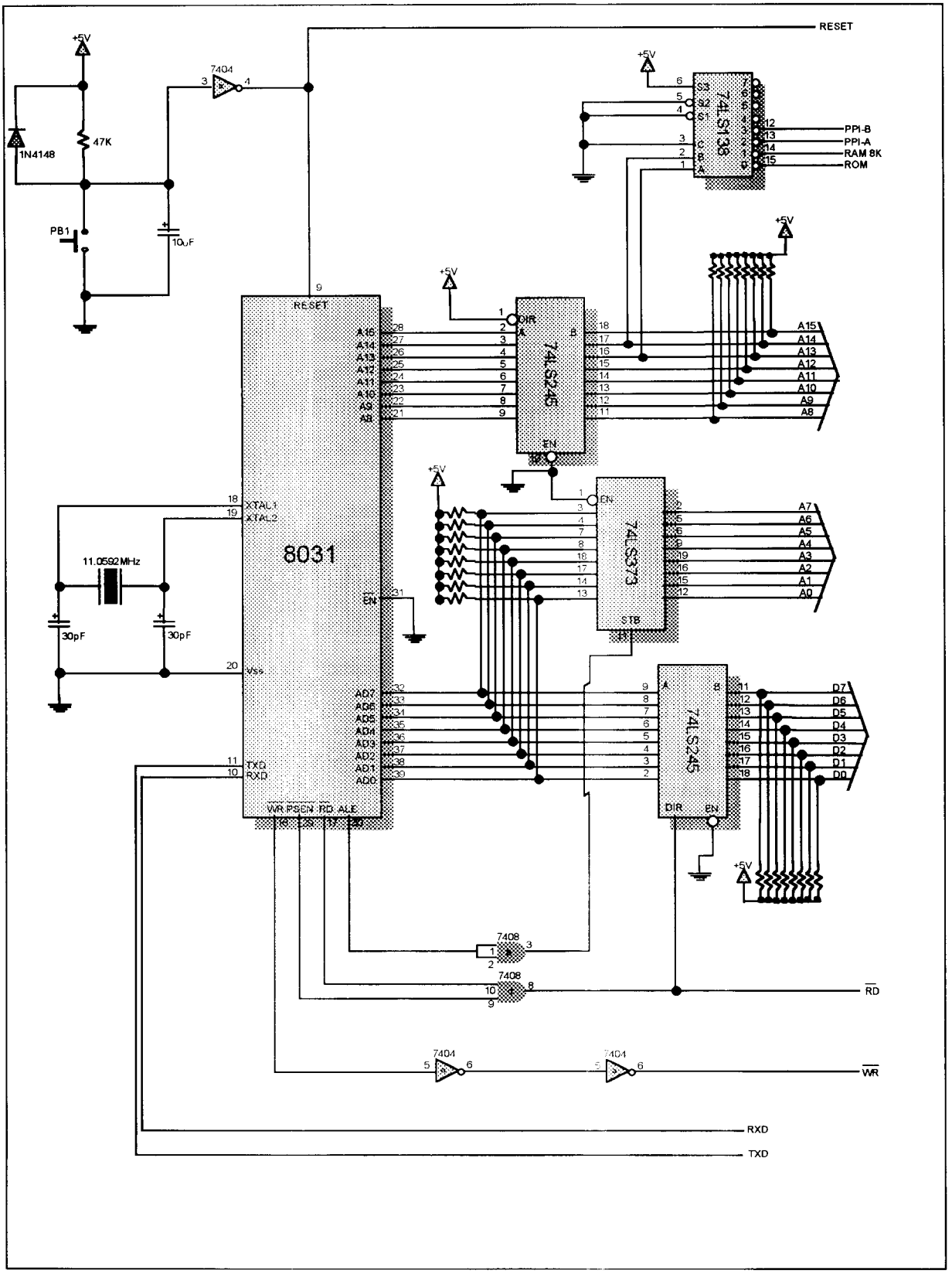


FIGURA 10. CONEXIONES PARA LA RAM, LA EPROM Y LAS PPIs

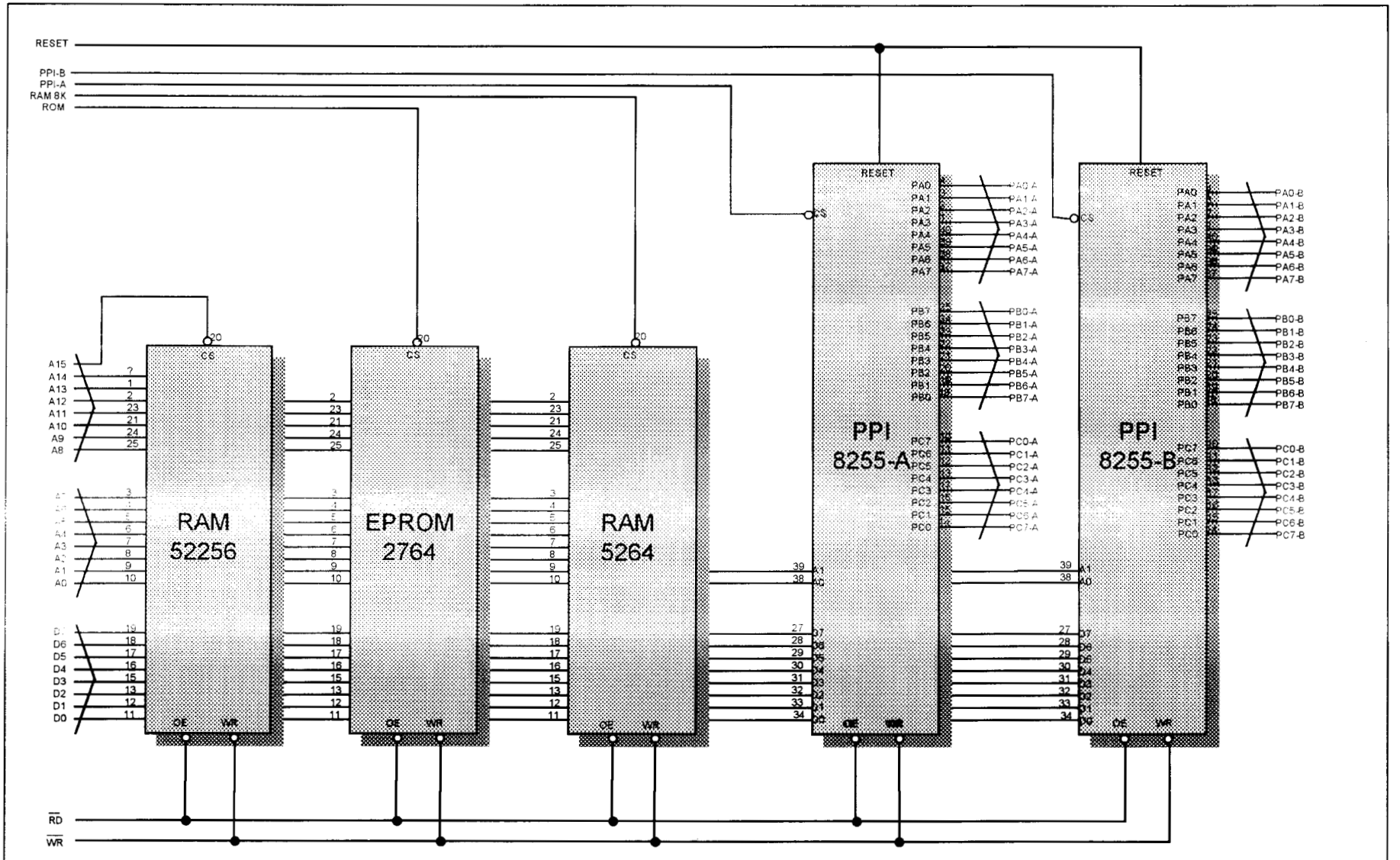
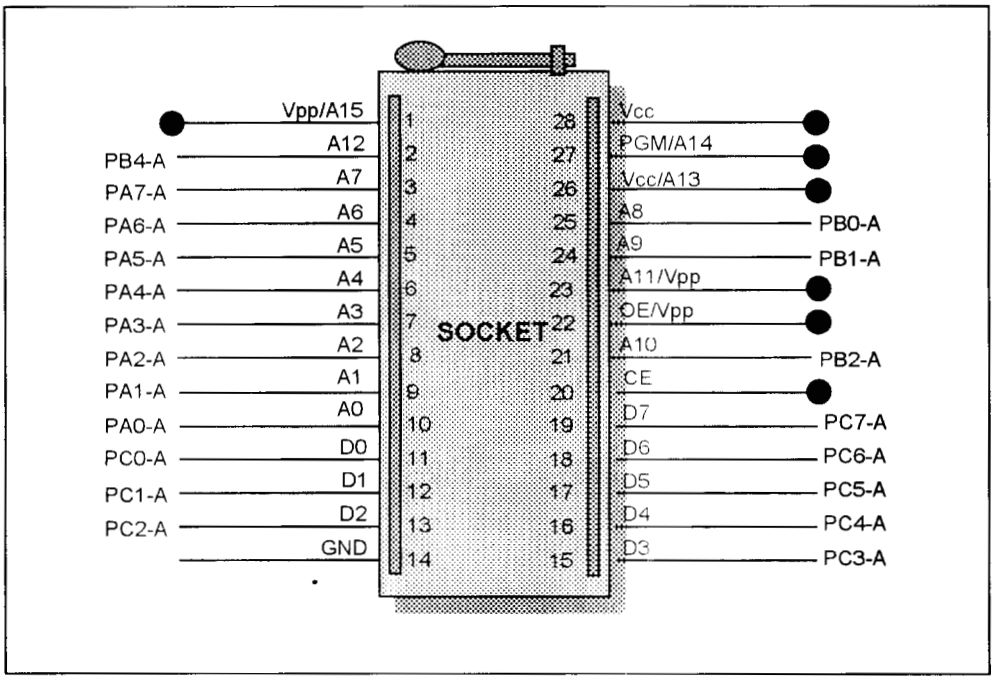


FIGURA 11. SOCKET DEL GRABADOR



NOTA: Los pines marcados con "● " son los pines que van a estar **controlados por los circuitos de las figuras** 1 y 3.

ANÁLISIS Y DISEÑO DEL SOFTWARE.

Para obtener un buen diseño del software, tenemos que analizar varias cosas primero, como son:

1.- Forma en la que se va a utilizar el grabador (Dependiente o Independiente)
2.- Determinar las funciones que tendrá el grabador, en base a el modo de trabajo elegido.

> En el caso de que sea de modo autónomo, se podrán realizar las operaciones de GRABAR y COMPARAR.
>
> En el caso de que sea en modo dependiente, se podrán realizar las operaciones de CARGAR DE PC A BUFFER, CARGAR DE BUFFER A PC, COPIAR BUFFER A EPROM, COPIAR EPROM A BUFFER, COMPARAR EPROM CON BUFFER, VERIFICAR SI EPROM BORRADA y SELECCIONAR TIPO, las cuales ya han sido descritas anteriormente.

3.- Implementar cada una de las funciones en base al modo de trabajo elegido.

Antes de la implantación del software, se diseñaron algunos diagramas de flujo principales, que nos marcan la forma en la que tienen que funcionar los programas.

En el diagrama de flujo 1 se muestra lo que sería el menú principal, con el cual podemos elegir el modo de trabajo. Notamos claramente lo que se menciono acerca de las funciones de los botones PB1 y PB2, donde PB2 nos sirve para desplazarnos en los menús y PB1 para aceptar la opción escogida por PB2. Al inicio se espera a que se pulse PB2 para iniciar, después se desplaza a modo "Independiente y modo "d"ependiente en forma alternada mientras se este presionando PB2. Si estando en "d" se presiona PB1 entonces nos iremos al menú del modo dependiente y si estamos en "I" y se presiona PB1 nos iremos al menú del modo independiente.

En el diagrama de flujo 2, observamos el menú del modo de trabajo independiente, donde solo se tienen las opciones de GRABAR DE EPROM A EPROM , de COMPARAR EL CONTENIDO DE LA EPROM CON EL CONTENIDO DEL BUFFER y la opción de salir que se maneja con una c minúscula.

El menú para modo de trabajo dependiente se muestra en el diagrama de flujo 3, en este modo se tienen algunas otras opciones además de las que permite el modo independiente. En este modo ya no se hace uso de los botones para manipular el flujo de la información, ahora se manipula el programa por medio de señales (números) que le manda un programa que estará corriendo en la PC (ITFC.EXE) a la que esté conectado el grabador por medio del puerto serie, entonces el programa está monitoreando el puerto serial y ejecutará la opción de acuerdo al caracter leído.

El diagrama de flujo 4 explica la rutina de GRABAR en modo independiente.

DIAGRAMA DE FLUJO 1. MENÚ PRINCIPAL

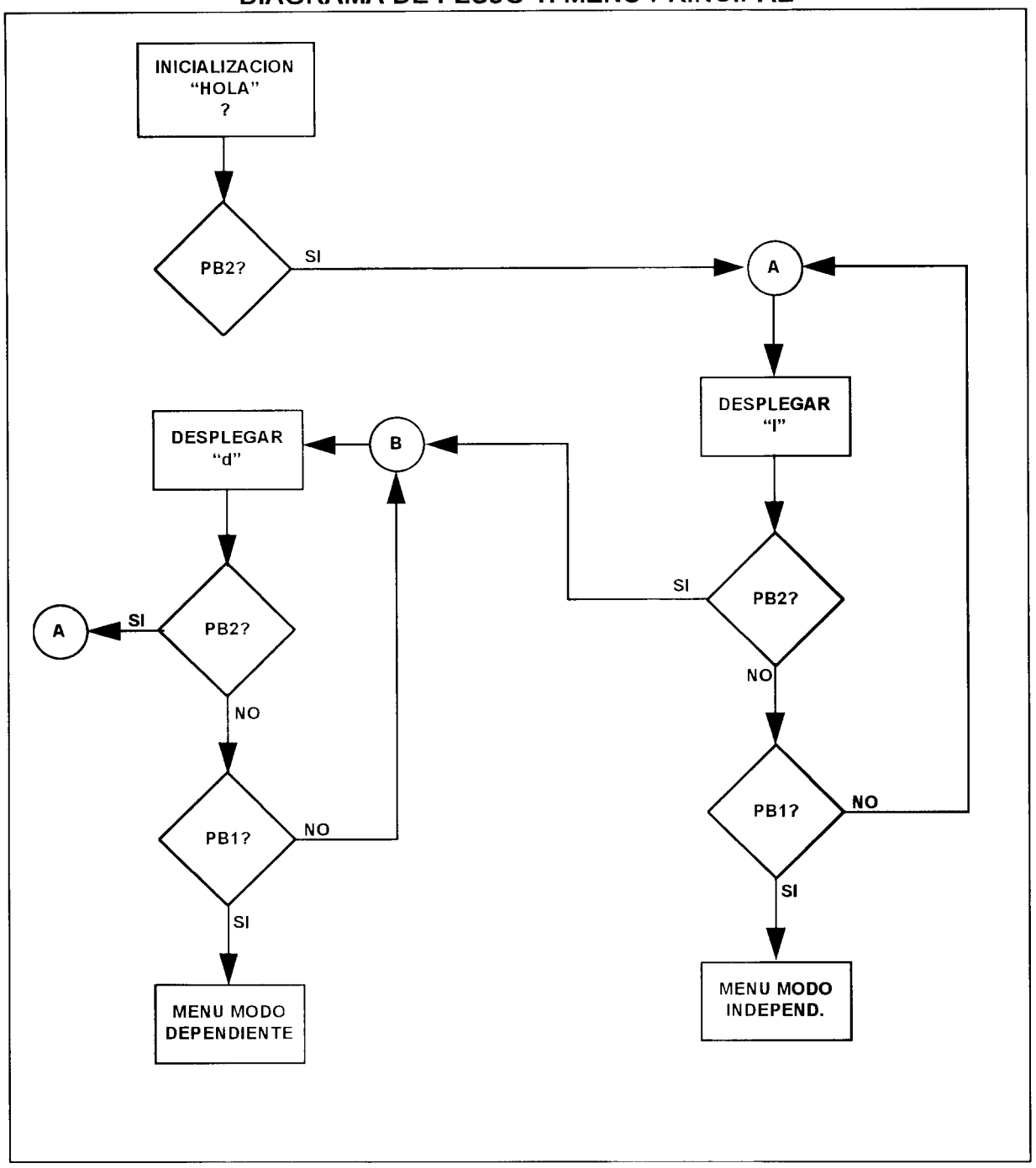


DIAGRAMA DE FLUJO 2. MENÚ EN MODO INDEPENDIENTE

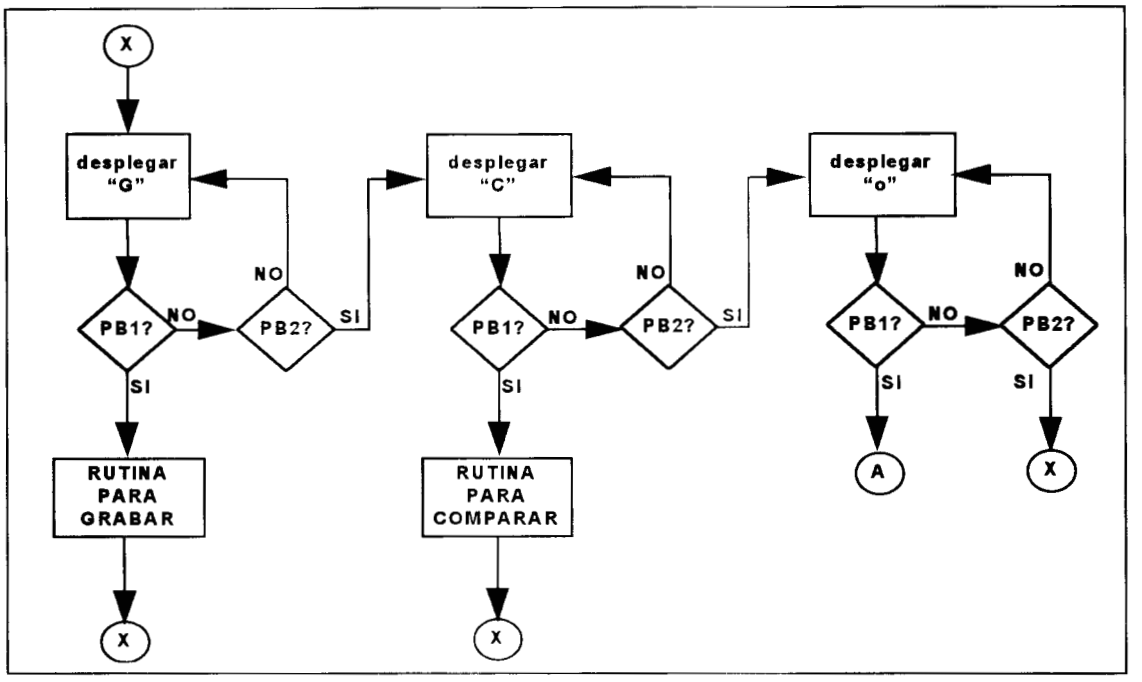


DIAGRAMA DE FLUJO 3. MENÚ EN MODO DEPENDIENTE

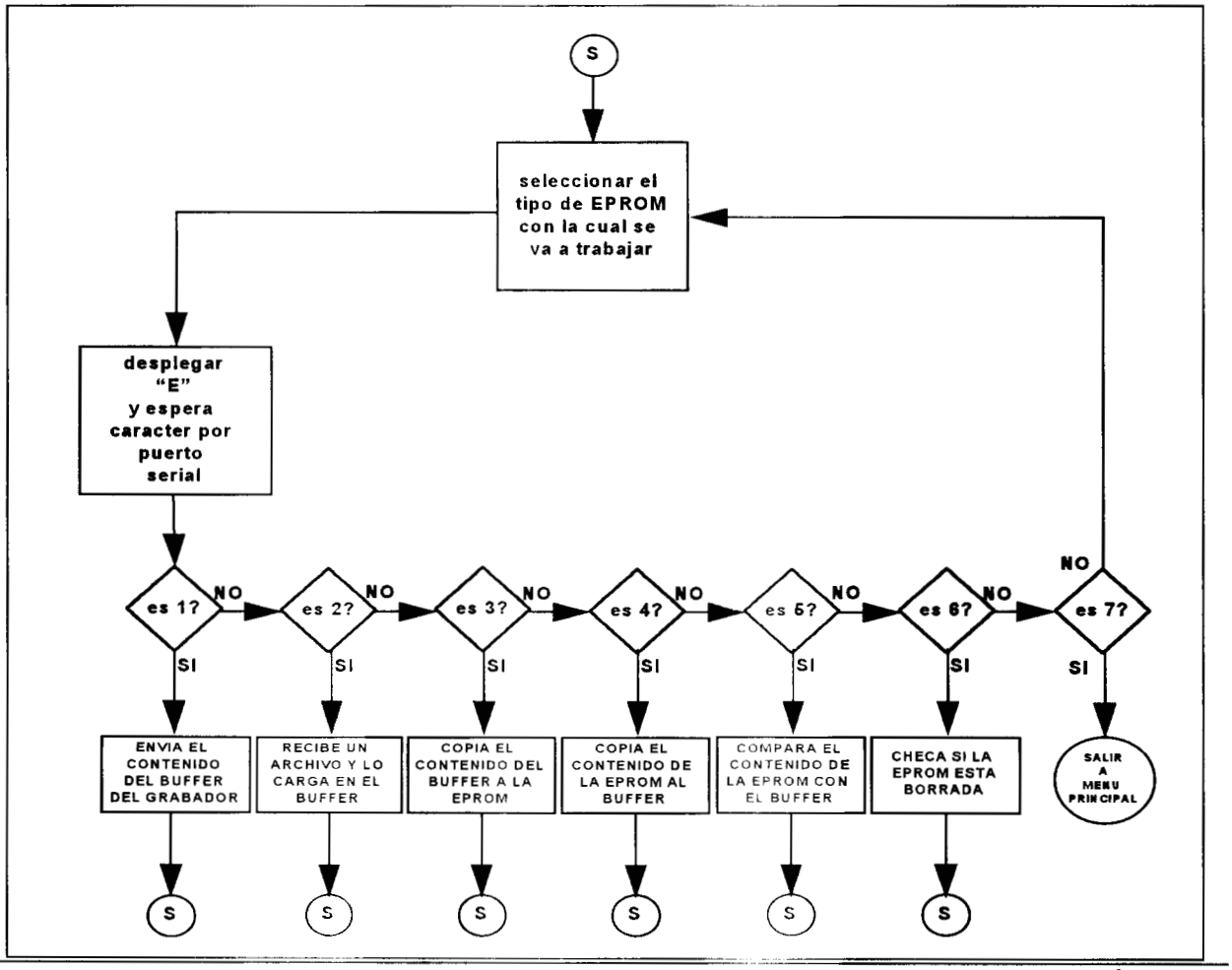
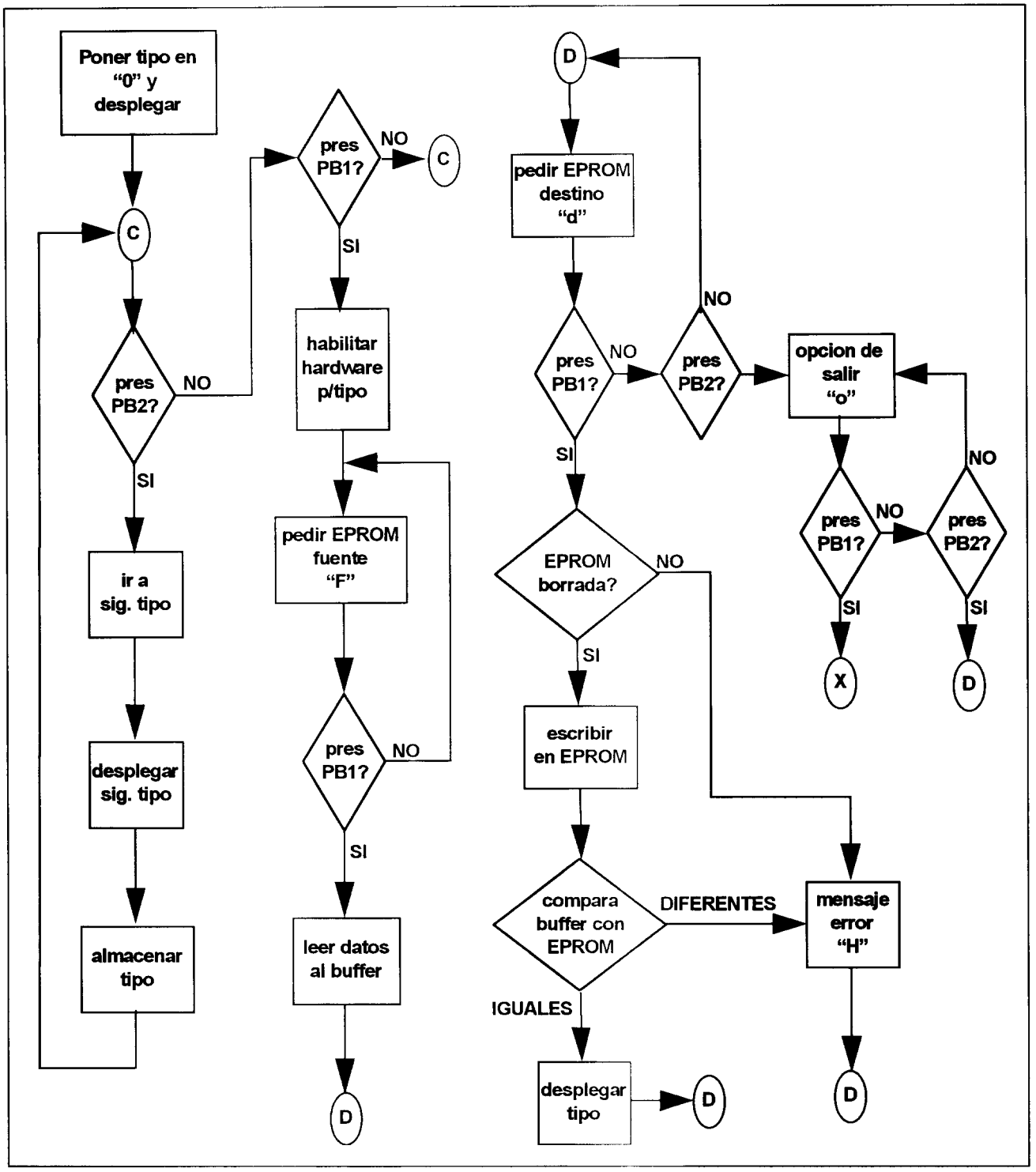


DIAGRAMA DE FLUJO 4. RUTINA PARA GRABAR EN MODO INDEPENDIENTE



Las rutinas de grabado para cada uno de los tipos de EPROMs permitidos en este grabador, se obtuvieron de algunos diagramas de flujo propuestos en el manual de INTEL de MEMORY, en el diagrama de flujo 5 se muestra un algoritmo de grabación QUICK PULSE, el cual funciona de la siguiente manera:

> Seleccionar dirección para grabar, poner voltajes de programación, poner datos a grabarse, inicializar una variable en 0, se da un pulso de programación de 100 microsegundos.
>
> Se incrementa la variable, se lee el dato grabado en la EPROM, se verifica que se haya grabado correctamente, si es así, se monitorea si es la última dirección y en caso positivo se ponen todos los voltajes en 5 volts y se realiza una comparación completa.
>
> En caso de que no se haya grabado correctamente se checa si la variable es igual a 25, si es igual a 25 se verifica el byte, si es correcto se checa si es la última dirección y se procede de la forma anterior, si no es correcto, se manda un mensaje de error.
>
> Cuando la variable es menor que 25 se vuelve a dar un pulso de programación de 100 microsegundos y se procede de la misma manera.

En el diagrama de flujo 6 se muestra un algoritmo de programación INTELIGENTE. La forma de operar de este algoritmo es similar al QUIK-PULSE, excepto por que manda un pulso extra de duración 3 veces el valor de la variable tratando de asegurar que la grabación sea correcta y para disminuir el tiempo de grabación.

Para cada tipo de EPROM las direcciones de fin varían de acuerdo a la capacidad y los voltajes de programación también.

Se recomienda el uso del algoritmo de programación QUICK-PULSE para aquellos tipos de EPROMs que usen voltajes de 21V y de 25V, y del algoritmo INTELIGENTE para los tipos que usen voltajes de 12.5 V.

DIAGRAMA DE FLUJO 5. Algoritmo de programación QUICK-PULSE

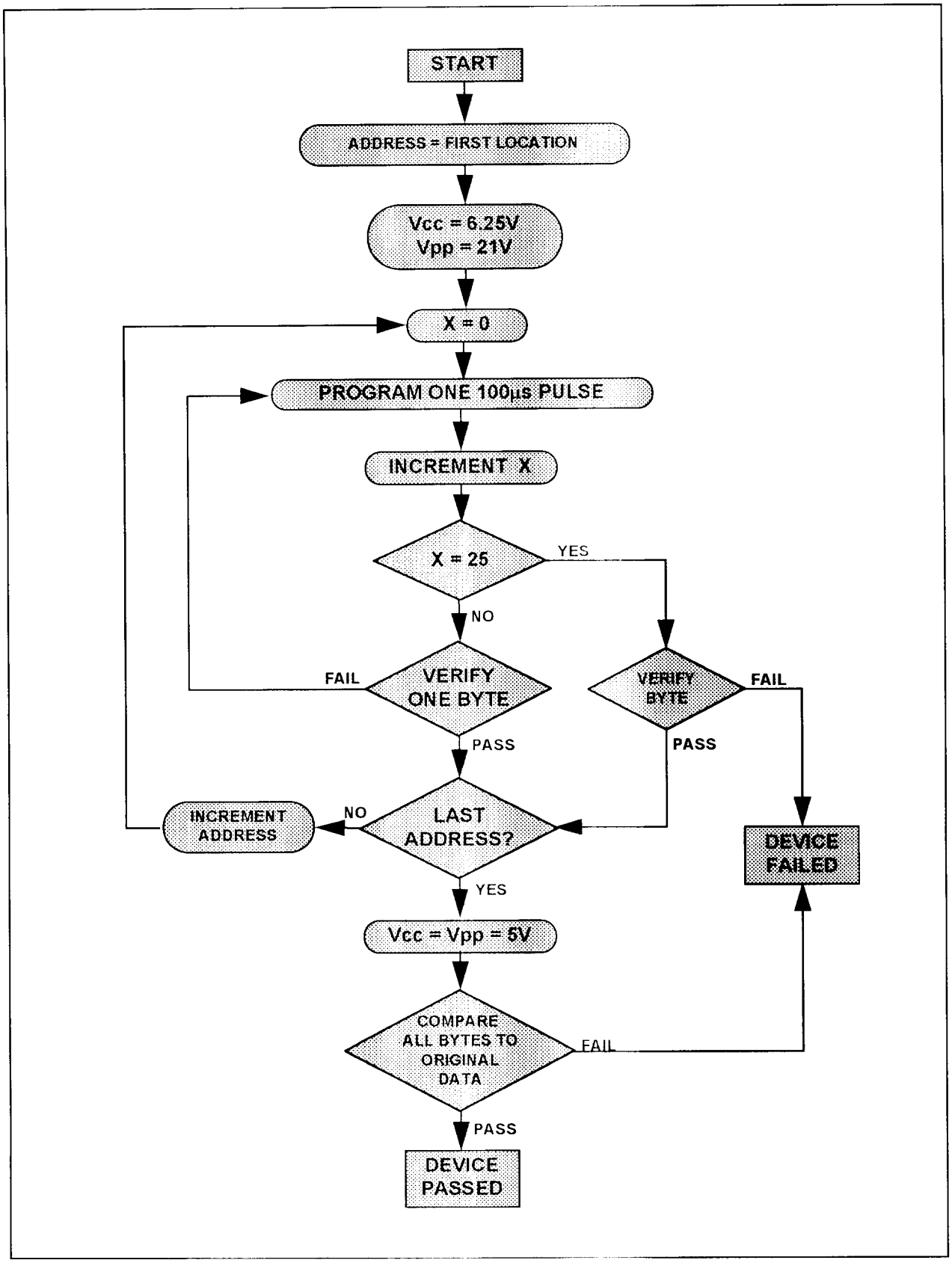
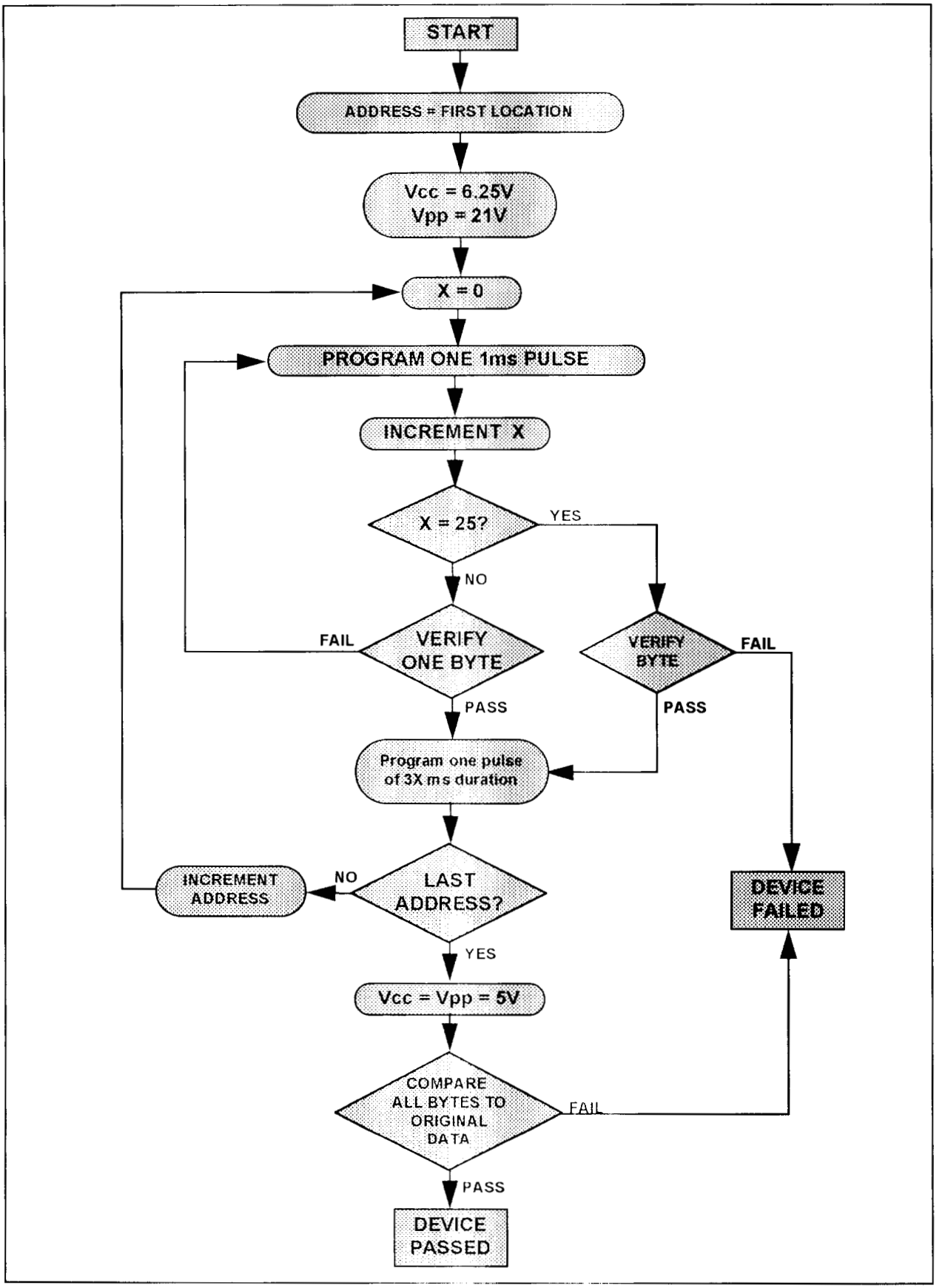


DIAGRAMA DE FLUJO 6. Algoritmo de programación INTELIGENTE



IMPLEMENTACIÓN DEL SOFTWARE.

A continuación se presenta el código fuente del programa **principal implementado únicamente para EPROMs 2764A.** En el ANEXO 2 se incluyen **listados de programas que fueron usados para probar el** sistema por partes.

```
0001  0000        ;*****************************************************************
0002  0000        ;LISTA DE ETIQUETAS
0003  0000            #INCLUDE EQUS.TXT
0110+ 0000        .LIST
0004  0000            .LIST
0005  0000
0006  0000        PCTRL2          .EQU    6003H     ;PALABRAS DE DIRECCION DE ACCESO PARA
0007  0000        PCTRL1          .EQU    4003H     ;CONFIGURAR LAS PPIs
0008  0000        PA2             .EQU    6000H     ;PALABRAS DE DIRECCION PARA ACCESAR A
0009  0000        PA1             .EQU    4000H     ;LOS PUERTOS DE CADA PPI
0010  0000        PB2             .EQU    6001H     ;SE USARA '1' PARA LA PPI-A
0011  0000        PB1             .EQU    4001H     ;Y '2' PARA LA PPI-B
0012  0000        PC2             .EQU    6002H
0013  0000        PC1             .EQU    4002H
0014  0000        DIRBUFF         .EQU    8000H     ;VARIABLES USADAS EN EL PROGRAMA
0015  0000        TIPO            .EQU    29H
0016  0000        TIPO.3          .EQU    30H
0017  0000        STAT0.0         .EQU    00H
0018  0000        STAT0.1         .EQU    01H
0019  0000        ADR0            .EQU    31H
0020  0000        ADR1            .EQU    32H
0021  0000        START0          .EQU    33H
0022  0000        START1          .EQU    34H
0023  0000        END0            .EQU    35H
0024  0000        END1            .EQU    36H
0025  0000        TEMP1           .EQU    37H
0026  0000        FILLK           .EQU    38H
0027  0000        FILLKW          .EQU    39H
0028  0000        CHKSUM          .EQU    3AH
0029  0000        NBYTES          .EQU    3BH
0030  0000        RTYPE           .EQU    3CH
0031  0000        TEMP2           .EQU    3DH
0032  0000        VBYTE           .EQU    3EH
0033  0000        X               .EQU    3FH
0034  0000        XCERO           .EQU    0DFH      ;CARACTERES QUE SE que se
0035  0000        XUNO            .EQU    086H      ;despliegen en el display
0036  0000        XDOS            .EQU    0BBH
0037  0000        XTRES           .EQU    0AFH
0038  0000        XCUATRO         .EQU    0E6H
0039  0000        XCINCO          .EQU    0EDH
0040  0000        XSEIS           .EQU    0FDH
0041  0000        XSIETE          .EQU    087H
0042  0000        XOCHO           .EQU    0FFH
0043  0000        XNUEVE          .EQU    0EFH
0044  0000        XA              .EQU    077H
0045  0000        XB              .EQU    07CH
0046  0000        XCMAY           .EQU    059H
0047  0000        XCMIN           .EQU    03CH
0048  0000        XD              .EQU    03EH
0049  0000        XE              .EQU    079H
0050  0000        XF              .EQU    071H
0051  0000        XG              .EQU    06FH
0052  0000        XH              .EQU    076H
0053  0000        XI              .EQU    050H
0054  0000        XL              .EQU    058H
0055  0000        XS              .EQU    06DH
0056  0000        XU              .EQU    05EH
0057  0000        XY              .EQU    06EH
0058  0000        XINT            .EQU    0B3H
0059  0000
0060  0000        ;*****************************************************************
0061  0000
0062  2000                .ORG    2000H       ;DIRECCION DE INICIO DEL PROGRAMA
```

```
0063  2000
0064  2000 90 60 03  INICIO:    MOV   DPTR,#PCTRL2   ;CONFIGURACION DE LA PPI-B
0065  2003 74 88                MOV   A,#88H        ;PARA MODOS DE OPERACION
0066  2005 F0                   MOVX  @DPTR,A       ;VER HOJAS TECNICAS
0067  2006 90 40 03             MOV   DPTR, #PCTRL1  ;CONFIGURACION DE LA PPI-A
0068  2009 74 80                MOV   A,#80H
0069  200B F0                   MOVX  @DPTR,A
0070  200C 31 A7                ACALL PWRDWN
0071  200E        ;*************************************************************
0072  200E 71 29    REGRESO:    ACALL HOLA
0073  2010 74 B3                MOV   A,#XINT       ;DESPLIEGA SIGNO DE INTERROGACION
0074  2012 71 BE                ACALL IMPRIME
0075  2014 71 66    MON_PB2:    ACALL CHECK_PBS     ;MONITOREA PB2 PARA SELECCIONAR
0076  2016 B4 40 FB             CJNE  A,#040H,MON_PB2 ;EL MODO DE TRABAJO
0077  2019 80 00                SJMP  MOD_INDP
0078  201B        ;*************************************************************
0079  201B 74 50    MOD_INDP:   MOV   A,#XI         ;MONITOREA PB1 PARA COMENZAR CON
0080  201D 71 BE                ACALL IMPRIME       ;MODO INDEPENDIENTE, O PB2 PARA
0081  201F 71 66    CHK1:       ACALL CHECK_PBS     ;CAMBIAR A MODO DEPENDIENTE
0082  2021 B4 40 02             CJNE  A,#040H,CHK2
0083  2024 01 2B                AJMP  MOD_DEP
0084  2026 B4 80 F6  CHK2       CJNE  A,#080H,CHK1
0085  2029 01 98                AJMP  MENU_IND
0086  202B        ;*************************************************************
0087  202B 74 3E    MOD_DEP:    MOV   A,#XD         ;MONITOREA PB1 PARA COMENZAR CON
0088  202D 71 BE                ACALL IMPRIME       MODO DEPENDIENTE, O PB2 PARA
0089  202F 71 66    CHECK1:     ACALL CHECK_PBS     ;CAMBIAR A MODO INDEPENDIENTE
0090  2031 B4 40 02             CJNE  A,#040H,CHECK2
0091  2034 01 1B                AJMP  MOD_INDP
0092  2036 B4 80 F6  CHECK2:    CJNE  A,#080H,CHECK1
0093  2039 01 3B                AJMP  MENU_DEP
0094  203B
0095  203B        ;*************************************************************
0096  203B 74 79    MENU_DEP    MOV  A,#XE
0097  203D 71 BE                ACALL IMPRIME
0098  203F 74 50                MOV  A,#50H     ;PROGRAMA PUERTO SERIE EN MODO 1
0099  2041 F5 98                MOV  SCON,A
0100  2043
0101  2043 74 F4                MOV  A,#0F4H    ;CARGA VALOR DE BAUDRATE
0102  2045 F5 8D                MOV  TH1,A
0103  2047
0104  2047 74 20                MOV  A,#20H     ;PROGRAMA TIMER 1 EN MODO 2
0105  2049 F5 89                MOV  TMOD,A
0106  204B
0107  204B 74 40                MOV  A,#40H     ;ARRANCA TIMER 1
0108  204D F5 88                MOV  TCON,A
0109  204F
0110  204F
0111  204F        ;*************************************************************
0112  204F        ;RUTINA DE MENU EN MODO DEPENDIENTE
0113  204F 90 60 02  WRRS  MOV    DPTR,#PC2
0114  2052 74 0F       MOV    A,#0FH
0115  2054 F0          MOVX   @DPTR,A
0116  2055 C0 E0       PUSH A
0117  2057 71 58       ACALL RECIBE
0118  2059 B4 01 04     CJNE A,#01,ESF2
0119  205C D0 E0       POP  A
0120  205E 01 6B       AJMP MEM_PROG
0121  2060 B4 02 04  ESF2  CJNE A,#02,ESSAL
0122  2063 D0 E0       POP  A
0123  2065 01 7D       AJMP PROG_MEM
0124  2067 D0 E0    ESSAL POP  A
0125  2069 01 2B       AJMP MOD_DEP
0126  206B        ;*************************************************************
0127  206B        ; MANDA 80 BYTES DEL CONTENIDO DEL BUFFER DE LA RAM DEL KIT A PC
0128  206B 71 58    MEM_PROG    ACALL RECIBE        ;RECIBE BYTE ALTO DE DPTR
0129  206D F5 83                MOV  DPH,A
0130  206F 71 58    WRDL        ACALL RECIBE        RECIBE BYTE BAJO DE DPTR
0131  2071 F5 82                MOV  DPL,A
0132  2073
0133  2073 79 80                MOV  R1,#80H
0134  2075
```

```
0135  2075 E0      LOOPE     MOVX A,@DPTR
0136  2076 71 4A              ACALL TRANS
0137  2078 A3                INC  DPTR          ;SIGUIENTE DATO
0138  2079 D9 FA              DJNZ R1,LOOPE
0139  207B 01 3B              AJMP MENU_DEP
0140  207D
0141  207D       ;*************************************************************************
0142  207D       ; RECIBE EL CONTENIDO DE UN PROGRAMA DE LA PC AL BUFFER DEL KIT
0143  207D 71 58   PROG_MEM   ACALL RECIBE     ;RECIBE EL TOTAL DE BYTES DE CODIGO
0144  207F FB                MOV  R3,A
0145  2080 71 58             ACALL RECIBE     ;R3 CONTIENE BYTE ALTO
0146  2082 FA                MOV  R2,A        ;R2 CONTIENE BYTE BAJO
0147  2083
0148  2083 71 58   LOOPCAP    ACALL RECIBE     ;RECIBE 2 BYTES DE DIRECCION
0149  2085 F5 83             MOV  DPH,A
0150  2087 71 58             ACALL RECIBE
0151  2089 F5 82             MOV  DPL,A
0152  208B 71 58             ACALL RECIBE     ;RECIBE DATO
0153  208D F0                MOVX @DPTR,A
0154  208E 1A                DEC  R2
0155  208F BA FF F1          CJNE R2,#0FFH,LOOPCAP
0156  2092 1B                DEC  R3
0157  2093 BB FF ED          CJNE R3,#0FFH,LOOPCAP
0158  2096 01 3B             AJMP  MENU_DEP ;WRRS
0159  2098
0160  2098       ;*************************************************************************
0161  2098       ;RUTINA DE MENU EN MODO INDEPENDIENTE
0162  2098 51 ED   MENU_IND    ACALL  CONF_ESCR
0163  209A 31 A7               ACALL  PWRDWN
0164  209C 74 6F               MOV    A,#XG
0165  209E 71 BE               ACALL  IMPRIME
0166  20A0 71 F6               ACALL  LED2
0167  20A2 31 A7               ACALL  PWRDWN
0168  20A4 71 66               ACALL  CHECK_PBS
0169  20A6 B4 40 1A            CJNE   A,#040H ;INICIO_GRAB
0170  20A9 74 59               MOV    A,#XCMAY
0171  20AB 71 BE               ACALL  IMPRIME
0172  20AD 71 F6               ACALL  LED2
0173  20AF 31 A7               ACALL  PWRDWN
0174  20B1 71 66               ACALL  CHECK_PBS
0175  20B3 B4 40 12            CJNE   A,#040H,INICIO_COMP
0176  20B6 74 3C               MOV    A,#XCMIN
0177  20B8 71 BE               ACALL  IMPRIME
0178  20BA 71 F6               ACALL  LED2
0179  20BC 31 A7               ACALL  PWRDWN
0180  20BE 71 66               ACALL  CHECK_PBS
0181  20C0 B4 40 0A            CJNE   A,#040H,SAL_IND        ;checa con pb2
0182  20C3
0183  20C3
0184  20C3 B4 80 D2  INICIO_GRAB   CJNE   A,#080H,MENU_IND
0185  20C6 01 D2                  AJMP   GRABAR_IND
0186  20C8 B4 80 CD  INICIO_COMP   CJNE   A,#080H,MENU_IND
0187  20CB 21 25                  AJMP   COMPR_IND
0188  20CD B4 80 C8  SAL_IND      CJNE   A,#080H,MENU_IND
0189  20D0 01 1B                  AJMP   MOD_INDP
0190  20D2
0191  20D2 75 29 FF  GRABAR_IND    MOV    TIPO,#0FFH
0192  20D5 05 29     PROX_TIPO:    INC    TIPO
0193  20D7 E5 29                   MOV    A,TIPO
0194  20D9 71 AB                   ACALL  GET7SC
0195  20DB 71 F6                   ACALL  LED2
0196  20DD 71 66                   ACALL  CHECK_PBS
0197  20DF B4 40 07                CJNE   A,#040H,START  ;PB2
0198  20E2 E5 29                   MOV    A,TIPO
0199  20E4 B4 08 EE                CJNE   A,#08H,PROX_TIPO
0200  20E7 01 D2                   AJMP   GRABAR_IND
0201  20E9
0202  20E9
0203  20E9 B4 80 E9  START:      CJNE   A,#080H,PROX_TIPO  ;PB1
0204  20EC 31 2B                  ACALL  INIC_BUFF  ;INICIALIZAR EL BUFFER DE LA RAM EN 00H
0205  20EE 74 71     FUENTE:     MOV    A,#XF
0206  20F0 71 BE                  ACALL  IMPRIME
```

```
0207  20F2 71 F6                    ACALL  LED2
0208  20F4 31 A7                    ACALL  PWRDWN
0209  20F6 71 66                    ACALL  CHECK_PBS
0210  20F8 B4 80 F3                 CJNE   A,#080H,FUENTE  ;PB1
0211  20FB 71 E3                    ACALL  LED1
0212  20FD 71 81                    ACALL  DELAYX
0213  20FF 31 4B                    ACALL  CCMD          ;CARGAR EL CONTENIDO DE LA EPROM EN LA RAM
0214  2101 74 3E     DESTINO:      MOV    A,#XD
0215  2103 71 BE                    ACALL  IMPRIME
0216  2105 31 A7                    ACALL  PWRDWN
0217  2107 71 66                    ACALL  CHECK_PBS
0218  2109 B4 80 0C                 CJNE   A,#080H,SALIR_DEST
0219  210C 71 E3                    ACALL  LED1
0220  210E 51 16                    ACALL  PCMD  ;PROGRAMAR LA EPROM CON EL CONTENIDO DE LA RAM
0221  2110 E5 29                    MOV    A,TIPO
0222  2112 71 AB                    ACALL  GET7SC
0223  2114 71 81                    ACALL  DELAYX
0224  2116 21 01                    AJMP   DESTINO
0225  2118 74 3C     SALIR_DEST    MOV    A,#XCMIN
0226  211A 71 BE                    ACALL  IMPRIME
0227  211C 31 A7                    ACALL  PWRDWN
0228  211E 71 66                    ACALL  CHECK_PBS
0229  2120 B4 80 DE                 CJNE   A,#80H,DESTINO
0230  2123 01 98                    AJMP   MENU_IND
0231  2125
0232  2125          ;********** INICIALIZAR BUFFER CON CEROS Y CARGAR EPROM A RAM **********
0233  2125 31 2B     COMPR_IND     ACALL INIC_BUFF
0234  2127 31 4B                   ACALL CCMD
0235  2129 01 98                   AJMP  MENU_IND
0236  212B
0237  212B          ;********** INICIALIZA EL BUFFER DEL KIT EN 00H **********
0238  212B
0239  212B C0 83     INIC_BUFF     PUSH   DPH
0240  212D C0 82                   PUSH   DPL
0241  212F C0 E0                   PUSH   A
0242  2131 90 30 00                MOV    DPTR,#3000H
0243  2134 74 00     PONCERO       MOV    A,#00H
0244  2136 F0                      MOVX   @DPTR,A
0245  2137 74 FE                   MOV    A,#0FEH
0246  2139 B5 82 0C                CJNE   A,DPL,PROX_CERO
0247  213C 74 3F                   MOV    A,#3FH
0248  213E B5 83 07                CJNE   A,DPH,PROX_CERO
0249  2141 D0 E0                   POP    A
0250  2143 D0 82                   POP    DPL
0251  2145 D0 83                   POP    DPH
0252  2147 22                      RET
0253  2148
0254  2148 A3       PROX_CERO     INC    DPTR
0255  2149 21 34                   AJMP   PONCERO
0256  214B
0257  214B          ;********** COPIA DATOS DE LA EPROM AL BUFER **********
0258  214B          ;LEE EL CONTENIDO DE UN RANGO DE LA EPROM ESPECIFICADO POR EL USUARIO
0259  214B          ;Y ES PUESTO EN LA MEMORIA DEL BUFFER USANDO LA MISMA DIRECCION
0260  214B
0261  214B          CCMD:         ;MOV DPTR,#PCTRL1
0262  214B                        ;MOV A,#89H
0263  214B                        ;MOVX @DPTR,A
0264  214B 31 74                   ACALL  PWRUP
0265  214D 71 00                   ACALL CONF_LECT
0266  214F 90 00 00   CCMD1:       MOV    DPTR,#0000H
0267  2152
0268  2152 31 C6     CREAD:       ACALL  RDPROM
0269  2154 C0 E0                   PUSH   A
0270  2156 E5 83                   MOV    A,DPH
0271  2158 44 30                   ORL    A,#00110000B
0272  215A F5 83                   MOV    DPH,A
0273  215C D0 E0                   POP    A
0274  215E F0                      MOVX   @DPTR,A
0275  215F 74 FF                   MOV    A,#0FFH
0276  2161 B5 82 06                CJNE   A,DPL,CNEXT
0277  2164 74 3F                   MOV    A,#3FH
0278  2166 B5 83 01                CJNE   A,DPH,CNEXT
```

```
0279  2169 22                        RET
0280  216A
0281  216A A3       CNEXT:      INC   DPTR
0282  216B E5 83                    MOV   A,DPH
0283  216D 54 CF                    ANL   A,#11001111B
0284  216F F5 83                    MOV   DPH,A
0285  2171 21 52                    AJMP  CREAD
0286  2173 22                        RET
0287  2174          ;********** POWER UP **********************************
0288  2174          ;este procedimiento inicializa los voltajes que se mandan al socket en
0289  2174          ;5 volts
0290  2174 90 60 00  PWRUP:      MOV   DPTR,#PA2
0291  2177 74 80                    MOV   A,#80H
0292  2179 F0                       MOVX  @DPTR,A
0293  217A 90 40 01                 MOV   DPTR,#PB1
0294  217D 74 17                    MOV   A,#17H
0295  217F F0                       MOVX  @DPTR,A
0296  2180 90 60 01                 MOV   DPTR,#PB2
0297  2183 74 A6                    MOV   A,#0A6H
0298  2185 F0                       MOVX  @DPTR,A
0299  2186 90 60 02                 MOV   DPTR,#PC2
0300  2189 74 0E                    MOV   A,#0EH
0301  218B F0                       MOVX  @DPTR,A
0302  218C 75 90 FF                 MOV   P1,#0FFH
0303  218F 31 92                    ACALL DLY50
0304  2191 22                        RET
0305  2192          ;**********  RETARDO DE 50 milisegundos **********
0306  2192 7B 32     DLY50:      MOV   R3,#32H
0307  2194 31 99     DL50:       ACALL DELAY1
0308  2196 DB FC                    DJNZ  R3,DL50
0309  2198 22                        RET
0310  2199
0311  2199          ;**********  retardo de 1 milisegundo **********
0312  2199 74 00     DELAY1:     MOV   A,#0
0313  219B D5 E0 FD  DL1:        DJNZ  A,DL1
0314  219E D5 E0 FD  DL2:        DJNZ  A,DL2
0315  21A1 22                        RET
0316  21A2
0317  21A2          ;********** RETARDO DE 20 microsegundos **********
0318  21A2 7B 0A     DLY20:      MOV   R3,#0AH
0319  21A4 DB FE     DLY201      DJNZ  R3,DLY201
0320  21A6 22                        RET
0321  21A7
0322  21A7          ;********** POWER DOWN **********************************
0323  21A7          ;aqui se mandan voltajes de 0 volts asi como datos y direcciones tambien 0
0324  21A7 90 40 00  PWRDWN:     MOV   DPTR,#PA1
0325  21AA 74 00                    MOV   A,#00H
0326  21AC F0                       MOVX  @DPTR,A
0327  21AD 90 60 02                 MOV   DPTR,#PC2
0328  21B0 74 0E                    MOV   A,#0EH
0329  21B2 F0                       MOVX  @DPTR,A
0330  21B3 90 60 01                 MOV   DPTR,#PB2
0331  21B6 74 A6                    MOV   A,#0A6H
0332  21B8 F0                       MOVX  @DPTR,A
0333  21B9 90 40 01                 MOV   DPTR,#PB1
0334  21BC 74 A8                    MOV   A,#0A8H
0335  21BE F0                       MOVX  @DPTR,A
0336  21BF 90 60 00                 MOV   DPTR,#PA2
0337  21C2 74 4E                    MOV   A,#4EH
0338  21C4 F0                       MOVX  @DPTR,A
0339  21C5 22                        RET
0340  21C6
0341  21C6          ;**********  RUTINA PARA LEER EPROM **********
0342  21C6          ;ESTA RUTINA LEE EL CONTENIDO DE UNA LOCALIDAD.
0343  21C6          ;EL 'DPTR' CONTIENE LA LOCALIDAD A SER LEIDA
0344  21C6          ;EL RESULTADO ES REGRESADO EN EL ACC
0345  21C6
0346  21C6 85 83 32  RDPROM:     MOV ADR1,DPH
0347  21C9 85 82 31                 MOV ADR0,DPL
0348  21CC
0349  21CC 21 CE                    AJMP RD64
0350  21CE
```

```
0351  21CE
0352  21CE          ;********************* EPROM 27x4(A)*********************************
0353  21CE E5 32    RD64:       MOV A,ADR1
0354  21D0 D2 E5                SETB A.5
0355  21D2 20 E3 0A             JB  A.3,A11H64
0356  21D5
0357  21D5 90 40 01  A11L64:    MOV DPTR,#PB1
0358  21D8 D2 E3                SETB A.3
0359  21DA 54 3F                ANL A,#00111111B
0360  21DC F0                   MOVX @DPTR,A
0361  21DD 21 E7                AJMP RDNX64
0362  21DF
0363  21DF 90 40 01  A11H64:    MOV DPTR,#PB3
0364  21E2 C2 E3                CLR A.3
0365  21E4 54 3F                ANL A,#00111111B
0366  21E6 F0                   MOVX @DPTR,A
0367  21E7
0368  21E7 90 40 00  RDNX64     MOV DPTR,#PA1
0369  21EA E5 31                MOV A,ADR0
0370  21EC F0                   MOVX @DPTR,A
0371  21ED 90 60 02             MOV DPTR,#PC2
0372  21F0 74 0E                MOV A,#0EH
0373  21F2 F0                   MOVX @DPTR,A
0374  21F3 90 60 01             MOV DPTR,#PB2
0375  21F6 74 26                MOV A,#26H
0376  21F8 F0                   MOVX @DPTR,A
0377  21F9 90 60 00             MOV DPTR,#PA2
0378  21FC 74 83                MOV A,#83H
0379  21FE F0                   MOVX @DPTR,A
0380  21FF 90 40 02             MOV DPTR,#PC1
0381  2202 31 A2                ACALL DLY20
0382  2204 E0                   MOVX A,@DPTR
0383  2205 C0 E0                PUSH A
0384  2207 90 60 00             MOV DPTR,#PA2
0385  220A 74 81                MOV A,#81H
0386  220C F0                   MOVX @DPTR,A
0387  220D D0 E0                POP A
0388  220F 85 32 83             MOV DPH,ADR1
0389  2212 85 31 82             MOV DPL,ADR0
0390  2215 22                   RET
0391  2216
0392  2216
0393  2216          ;****************** ESCRIBE DATOS DEL BUFFER A LA EPROM ******************
0394  2216          ;PROGRAMA EL SOCKET SEGUN LA EPROM
0395  2216 51 ED    PCMD:       ACALL CONF_ESCR
0396  2218 31 74                ACALL PWRUP
0397  221A 75 33 00  PCMD1:     MOV  START0,#00H
0398  221D 75 34 00             MOV  START1,#00H
0399  2220 85 33 82             MOV  DPL,START0
0400  2223 85 34 83             MOV  DPH,START1
0401  2226
0402  2226 51 ED    PAGAIN:     ACALL CONF_ESCR
0403  2228 71 13                ACALL TDPE
0404  222A E0                   MOVX  A,@DPTR
0405  222B 71 1E                ACALL TDPE
0406  222D 51 59                ACALL WREPROM
0407  222F 20 01 0D             JB   STATC,PERR
0408  2232 74 FF                MOV  A,#0FFH
0409  2234 B5 82 1E             CJNE  A,DPL,PNEXT
0410  2237 74 0F                MOV  A,#0FH
0411  2239 B5 83 19             CJNE  A,DPH,PNEXT
0412  223C 31 A7                ACALL PWRDWN
0413  223E 22                   RET
0414  223F
0415  223F 74 76    PERR:       MOV  A,#XH
0416  2241 71 BE                ACALL IMPRIME
0417  2243 71 81                ACALL DELAYX
0418  2245 71 81                ACALL DELAYX
0419  2247 74 86                MOV  A,#XLOW
0420  2249 71 BE                ACALL IMPRIME
0421  224B 71 81                ACALL DELAYX
0422  224D 71 81                ACALL DELAYX
```

```
0423  224F 71 66                    ACALL  CHE  PBS
0424  2251 B4 80 EB                 CJNE   A,#0  PERR
0425  2254 22                       RET
0426  2255
0427  2255 A3       PNEXT:         INC    DPTR
0428  2256 41 26                   AJMP   PAG
0429  2258 22                       RET
0430  2259
0431  2259          ;************* RUTINA PARA ESC  R EPROM          ****************
0432  2259          ;ESTA RUTINA ESCRIBE EL CON  NIDO DEL BUFFER A UNA LOCALIDAD.
0433  2259          ;EL 'DPTR' CONTIENE LA LOCAL  AD A SER ESCRITA
0434  2259          ;EL VALOR A ESCRIBIRSE ESTA  EL ACC
0435  2259
0436  2259 85 83 32  WRPROM:       MOV ADR1,D
0437  225C 85 82 31                MOV ADR0,DP
0438  225F F5 39                   MOV FILLKW
0439  2261
0440  2261 41 63                   AJMP WR64A
0441  2263
0442  2263          ;********************* EPROM 27C  ***************************************
0443  2263 90 60 00  WR64A:        MOV  DPTR,#P
0444  2266 74 81                   MOV  A,#081
0445  2268 F0                      MOVX @DPT  A
0446  2269 90 60 01                MOV  DPTR,#  2
0447  226C 74 2D                   MOV  A,#2D
0448  226E F0                      MOVX @DP  A
0449  226F 90 60 02                MOV  DPTR,#  2
0450  2272 74 0E                   MOV  A,#0EH
0451  2274 F0                      MOVX @DPT  A
0452  2275 74 32                   MOV  A,#ADP
0453  2277 D2 E5                   SETB A.5
0454  2279 20 E3 0A                JB   A.3,W
0455  227C
0456  227C 90 40 01  W11L6A:       MOV  DPTR,#P
0457  227F D2 E3                   SETB A.3
0458  2281 54 3F                   ANL  A,#00  11B
0459  2283 F0                      MOVX @DP  A
0460  2284 41 8E                   AJMP  WNX6
0461  2286
0462  2286 90 40 01  W11H6A:       MOV  DPTR,#P
0463  2289 C2 E3                   CLR  A.3
0464  228B 54 3F                   ANL  A,#001  11B
0465  228D F0                      MOVX @DPTR,A
0466  228E
0467  228E 90 40 00  WNX64A:       MOV  DPTR,#P
0468  2291 E5 31                   MOV  A,ADR
0469  2293 F0                      MOVX @DP  A
0470  2294 90 40 02                MOV  DPTR,#  1
0471  2297 31 A2                   ACALL DLY
0472  2299 31 A2                   ACALL DLY
0473  229B 31 A2                   ACALL DLY
0474  229D 31 A2                   ACALL DLY
0475  229F 75 3F 00                MOV  X,#0
0476  22A2 E5 39     AGN64:        MOV  A,FILLKW
0477  22A4 F0                      MOVX @DP  A
0478  22A5 90 60 00                MOV  DPTR,#  2
0479  22A8 74 85                   MOV  A,#85H
0480  22AA F0                      MOVX @DP  A
0481  22AB C0 E0                   PUSH A
0482  22AD 31 A2                   ACALL DLY
0483  22AF 31 A2                   ACALL DLY
0484  22B1 31 A2                   ACALL DLY
0485  22B3 31 A2                   ACALL DLY
0486  22B5 D0 E0                   POP  A
0487  22B7 05 3F                   INC  X
0488  22B9
0489  22B9 E5 3F     NOV64A:       MOV  A,X
0490  22BB C0 E0                   PUSH A
0491  22BD 90 60 00                MOV  DPTR,#  2
0492  22C0 74 81                   MOV  A,#81
0493  22C2 F0                      MOVX @DP  A
0494  22C3 90 40 02                MOV DPTR,#  1
```

```
0495  22C6 D0 E0                    POP  A
0496  22C8 B4 20 D7                 CJNE A,#20H,AGN64
0497  22CB
0498  22CB E5 3F      OVER64:       MOV  A,X
0499  22CD 90 60 00                 MOV  DPTR,#6000H
0500  22D0 74 85                    MOV  A,#85H
0501  22D2 F0                       MOVX @DPTR,A
0502  22D3 90 60 00                 MOV  DPTR,#6000H
0503  22D6 74 81                    MOV  A,#81H
0504  22D8 F0                       MOVX @DPTR,A
0505  22D9 C2 01                    CLR  STATC
0506  22DB 85 32 83                 MOV  DPH,ADDR1
0507  22DE 85 31 82                 MOV  DPL,ADDR0
0508  22E1 22                       RET
0509  22E2 41 E4      PERR6A:       AJMP PGMERR
0510  22E4
0511  22E4          ;************** RUTINA DE ERROR EN PROGRAMACION **********************
0512  22E4 D2 01      PGMERR        SETB STATC
0513  22E6 85 32 83                 MOV  DPH,ADDR1
0514  22E9 85 31 82                 MOV  DPL,ADDR0
0515  22EC 22                       RET
0516  22ED
0517  22ED          ;************** CONFIG. PPI-A PARA ESCRITURA ************************
0518  22ED          ;en especial puerto C de la PPI-A
0519  22ED C0 E0      CONF_ESCR     PUSH A
0520  22EF C0 83                    PUSH DPH
0521  22F1 C0 82                    PUSH DPL
0522  22F3 90 40 03                 MOV  DPTR,#PCTRL1
0523  22F6 74 80                    MOV  A,#80H
0524  22F8 F0                       MOVX @DPTR,A
0525  22F9 D0 82                    POP DPL
0526  22FB D0 83                    POP DPH
0527  22FD D0 E0                    POP A
0528  22FF 22                       RET
0529  2300
0530  2300          ;************** CONFIG. PPI-A PARA LECTURA **************************
0531  2300          ;en especial puerto C de la PPI-A
0532  2300 C0 E0      CONF_LECT     PUSH A
0533  2302 C0 83                    PUSH DPH
0534  2304 C0 82                    PUSH DPL
0535  2306 90 40 03                 MOV  DPTR,#PCTRL1
0536  2309 74 89                    MOV  A,#89H
0537  230B F0                       MOVX @DPTR,A
0538  230C D0 82                    POP DPL
0539  230E D0 83                    POP DPH
0540  2310 D0 E0                    POP A
0541  2312 22                       RET
0542  2313
0543  2313          ;************** TRANSFORMA DIR. DE EPROM A RAM ***********************
0544  2313          ;para efectos de grabar apartir de la localidad 3000h
0545  2313 C0 E0      TDPB          PUSH A
0546  2315 E5 83                    MOV  A,DPH
0547  2317 44 30                    ORL  A,#00110000B
0548  2319 F5 83                    MOV  DPH,A
0549  231B D0 E0                    POP A
0550  231D 22                       RET
0551  231E
0552  231E          ;************** TRANSFORMA DIR. DE RAM A EPROM ***********************
0553  231E          ;para grabar apartir de la direccion 2000h de la EPROM destino
0554  231E C0 E0      TDPE          PUSH A
0555  2320 E5 83                    MOV  A,DPH
0556  2322 54 CF                    ANL  A,#11001111B
0557  2324 F5 83                    MOV  DPH,A
0558  2326 D0 E0                    POP A
0559  2328 22                       RET
0560  2329
0561  2329
0562  2329          ;********************* IMPRIME "HOLA" ******************************
0563  2329 74 76      HOLA          MOV  A,#76H
0564  232B 71 BE                    ACALL IMPRIME
0565  232D 71 81                    ACALL DELAY
0566  232F 71 81                    ACALL DELAY
```

```
0567 2331 74 5F                  MOV A,#5F
0568 2333 71 BE                  ACALL IMPR_1E
0569 2335 71 81                  ACALL DEL_X
0570 2337 71 81                  ACALL DELA_X
0571 2339 74 58                  MOV A,#58H
0572 233B 71 BE                  ACALL IMPR_1E
0573 233D 71 81                  ACALL DEL_X
0574 233F 71 81                  ACALL DELA_X
0575 2341 74 77                  MOV A,#77H
0576 2343 71 BE                  ACALL IMPR_1E
0577 2345 71 81                  ACALL DELA_X
0578 2347 71 81                  ACALL DELA_X
0579 2349 22                     RET
0580 234A        ;*****************************************************************
0581 234A F5 99    TRANS: MOV SBUF,A
0582 234C C0 E0           PUSH   A
0583 234E 71 95           ACALL DISPLAY
0584 2350 D0 E0           POP    A
0585 2352 30 99 FD  WTB:   JNB SCON.1,WTB      TRANSMITE BYTE DIRECCIONADO
0586 2355 C2 99           CLR SCON.1
0587 2357 22              RET
0588 2358        ;*****************************************************************
0589 2358 30 98 FD  RECIBE: JNB SCON.0,RECIBE      ;RECIBE ECO DE TERMINAL
0590 235B C2 98           CLR SCON.0
0591 235D E5 99           MOV A,SBUF
0592 235F C0 E0           PUSH A
0593 2361 71 95           ACALL DISPLAY
0594 2363 D0 E0           POP  A
0595 2365 22              RET
0596 2366
0597 2366        ;*****************************************************************
0598 2366        ;monitorea los botones de control (PB1 Y PB2) y no regresa hasta que no
0599 2366        ;se haya presionado uno de los 2
0600 2366 C0 83   CHECK_PBS:   PUSH   DPH
0601 2368 C0 82                PUSH   DPL
0602 236A 90 60 02             MOV    DPTR,#PC2
0603 236D 74 00   CHECA      MOV    A,#00H
0604 236F E0                   MOVX   A,@DPTR
0605 2370 54 C0                ANL    A,#0C0H
0606 2372 30 E6 05             JNB    ACC.6,FIN_PBS
0607 2375 30 E7 02             JNB    ACC.7,FIN_PBS
0608 2378 61 6D                AJMP   CHECA
0609 237A 71 81   FIN_PBS    ACALL  DELAYX
0610 237C D0 82                POP    DPL
0611 237E D0 83                POP    DPH
0612 2380 22                   RET
0613 2381
0614 2381        ;*****************************************************************
0615 2381
0616 2381 79 60   DELAYX     MOV  R1,#60H      ;RETARDO
0617 2383 D9 01   DELAY0     DJNZ R1,DELAY3
0618 2385 22                   RET
0619 2386
0620 2386 78 00   DELAY3     MOV  R0,#00H
0621 2388 D8 FE    RETA1     DJNZ R0,RETA1
0622 238A D8 FE    RETA2     DJNZ R0,RETA2
0623 238C D8 FE    RETA3     DJNZ R0,RETA3
0624 238E D8 FE    RETA4     DJNZ R0,RETA4
0625 2390 D8 FE    RETA5     DJNZ R0,RETA5
0626 2392 61 83                AJMP DELAY0
0627 2394 22                   RET
0628 2395
0629 2395        ;*****************************************************************
0630 2395        ; ESCRIBE EN EL DISPLAY EL CONTENIDO DEL ACUMULADOR EN DOS DIGITOS HEXA
0631 2395 C0 82   DISPLAY:   PUSH DPL      ESCRIBE EN EL DISPLAY EL CONTENIDO DEL
0632 2397 C0 83            PUSH DPH      ACUMULADOR EN DIGITOS HEXADECIMALES.
0633 2399 C0 E0            PUSH A
0634 239B 54 0F            ANL A,#0FH
0635 239D 71 AB            ACALL GET7SC    ;SE ESCRIBE EL BYTE BAJO
0636 239F D0 E0            POP A
0637 23A1 C4               SWAP A
0638 23A2 54 0F            ANL A,#0FH
```

```
0639  23A4 71 AB              ACALL GET7SC    ;SE ESCRIBE EL BYTE ALTO
0640  23A6 D0 83              POP DPH
0641  23A8 D0 82              POP DPL
0642  23AA 22                 RET
0643  23AB          ;*************************************************************************
0644  23AB          ;REGRESA EL CODIGO DE 7 SEGMENTOS DEL ACUMULADOR
0645  23AB
0646  23AB C0 82    GET7SC    PUSH DPL
0647  23AD C0 83              PUSH DPH
0648  23AF C0 F0              PUSH B
0649  23B1 90 24 09           MOV DPTR,#TABLA
0650  23B4 93                 MOVC A,@A+DPTR      ; GARGA DATO DE 7 SEGMENTOS
0651  23B5 71 BE              ACALL IMPRIME
0652  23B7 D0 F0              POP  B
0653  23B9 D0 83              POP  DPH
0654  23BB D0 82              POP  DPL
0655  23BD 22                 RET
0656  23BE
0657  23BE
0658  23BE          ;*************************************************************************
0659  23BE          ;manda al display el contenido del acumulador cargado antes de llamar
0660  23BE          ;a este procedimiento, normalmente con algun codigo establecido en la
0661  23BE          ;etiquetas
0662  23BE C0 82    IMPRIME:    PUSH  DPL
0663  23C0 C0 83                PUSH  DPH
0664  23C2 C0 E0                PUSH  A
0665  23C4 90 60 00             MOV   DPTR,#PA2
0666  23C7 78 08                MOV   R0,#08
0667  23C9 F5 F0    ETQ4:       MOV   B,A
0668  23CB 54 80                ANL   A,#80H
0669  23CD 03                   RR    A
0670  23CE 03                   RR    A
0671  23CF 44 10                ORL   A,#10H
0672  23D1 F0                   MOVX  @DPTR,A
0673  23D2 54 EF                ANL   A,#0EFH
0674  23D4 F0                   MOVX  @DPTR,A
0675  23D5 E5 F0                MOV   A,B
0676  23D7 23                   RL    A
0677  23D8 D8 EF                DJNZ  R0,ETQ4
0678  23DA 74 00                MOV   A,#00H
0679  23DC D0 E0                POP   A
0680  23DE D0 83                POP   DPH
0681  23E0 D0 82                POP   DPL
0682  23E2 22                   RET
0683  23E3          ;*************************************************************************
0684  23E3          ;rutina que permite encender al led 1 (led rojo)
0685  23E3 C0 E0    LED1:       PUSH  A
0686  23E5 C0 82                PUSH  DPL
0687  23E7 C0 83                PUSH  DPH
0688  23E9 90 60 00             MOV   DPTR,#PA2
0689  23EC 74 80                MOV   A,#80H
0690  23EE F0                   MOVX  @DPTR,A
0691  23EF D0 83                POP   DPH
0692  23F1 D0 82                POP   DPL
0693  23F3 D0 E0                POP   A
0694  23F5 22                   RET
0695  23F6          ;*************************************************************************
0696  23F6          ;rutina para encender led 2 (led verde)
0697  23F6 C0 E0    LED2:       PUSH  A
0698  23F8 C0 82                PUSH  DPL
0699  23FA C0 83                PUSH  DPH
0700  23FC 90 60 00             MOV   DPTR,#PA2
0701  23FF 74 40                MOV   A,#40H
0702  2401 F0                   MOVX  @DPTR,A
0703  2402 D0 83                POP   DPH
0704  2404 D0 82                POP   DPL
0705  2406 D0 E0                POP   A
0706  2408 22                   RET
0707  2409          ;*************************************************************************
0708  2409          ; TABLAS DE DATOS
0709  2409                      .gfedcba
0710  2409 DF       TABLA       BYTE   11011111B   0
```

```
0711  240A 86                .BYTE   10000110B   ;1
0712  240B BB                .BYTE   10111011B   ;2
0713  240C AF                .BYTE   10101111B   ;3
0714  240D E6                .BYTE   11100110B   ;4
0715  240E ED                .BYTE   11101101B   ;5
0716  240F FD                .BYTE   11111101B   ;6
0717  2410 87                .BYTE   10000111B   ;7
0718  2411 FF                .BYTE   11111111B   ;8
0719  2412 EF                .BYTE   11101111B   ;9
0720  2413 77                .BYTE   01110111B   ;A
0721  2414 7C                .BYTE   01111100B   ;B
0722  2415 59                .BYTE   01011001B   ;C
0723  2416 3E                .BYTE   00111110B   ;D
0724  2417 79                .BYTE   01111001B   ;E
0725  2418 71                .BYTE   01110001B   ;F
0726  2419
0727  2419
0728  241A            .ORG $+1        ;OBLIGA AL ENSAMBLADOR A GENERAR
0729  241A 00 20   RUN    .WORD   INICIO  ;POR SEPARADO UNA LINEA DE CODIGO
0730  241C            .END            ;PARA LA DIRECCION DE ARRANQUE.
0731  241C
0732  241C
0733  241C
tasm: Number of errors = 0
```

RESULTADOS

En relación a los avances obtenidos en el presente proyecto podemos establecer que:

Es posible "simular" la lectura y escritura al socket, es decir, se pueden enviar señales para datos, direcciones y control de voltajes tanto para leer como para escribir de la EPROM. Se dice "simular" por el hecho de que no se han logrado realizar dichas funciones en forma efectiva cuando se coloca una EPROM en el socket, únicamente se han verificado con una punta lógica y un multímetro.

Para los voltajes de control se sigue la siguiente técnica:

> Si se quiere tener un voltaje X en el pin Y del socket, se debe enviar un 0 lógico a la línea correspondiente al voltaje X del pin Y y 1's en las demás líneas de voltajes. Esta técnica es válida en los pines 1, 22, 23, 26 y 28.
>
> Por ejemplo si queremos que en el pin 22 se obtengan voltajes de 21 Volts y de 0 Volts respectivamente tendremos las siguientes asignaciones:
>
> a) PC0-B = 1, PC1-B = 1, PC2-B = 0, PC3-B = 1, PA1-B = 0.
> b) PC0-B = X, PC1-B = X, PC2-B = X, PC3-B = X, PA1-B = 1.
>
> donde la X significa cualquier valor (1 o 0).

Para los pines 20 y 27 funciona de la siguiente manera:

> El PB6-B para el pin 27 y el PB6-B para el pin 20 tienen la función de selectores, es decir, estas líneas establecen si la salida cambia respecto a una (PA2-B, si PB6-B esta en 1) u otra (PB6-A, si PB6-B esta en 0) línea de control (estos valores son para el pin 27 del socket, para el pin 20 funciona de la misma forma), el valor obtenido en la salida será la entrada seleccionada invertida.
>
> Por ejemplo si queremos que en el pin 27 se obtenga primero un 1 y después un 0 proporcionados por PA2-B tendremos las siguientes asignaciones:
>
> a) PB6-B = 1, PB6-A = X, PA2-B = 0.
> b) PB6-B = 1, PB6-A = X, PA2-B = 1.
>
> donde la X significa cualquier valor (1 o 0).

Las PPIs (8255) están programadas según las hojas técnicas (ver anexo 1), de la siguiente manera:

> PPI-A se programa con todos los puertos de salida en la fase de escritura de la EPROM en el socket, y se programa con los puertos A y B de salida y el puerto C de entrada en la fase de lectura de la EPROM en el socket, esto es por que el puerto C de la PPI-A maneja los datos.

PPI-B se programa con los puertos A, B y la parte alta del puerto C como salidas y la parte baja del puerto C como entrada, durante todo el desarrollo del programa, el hecho de que la parte baja del puerto C (PC0-PC3) se programe como entrada es por que son las líneas que controlan los botones (PB1 y PB2) y estos proporcionan siempre una entrada al sistema.

El sistema ya cuenta con el prototipo de todo el programa en el cual ya están establecidos todos los mensajes necesarios para la comunicación con el usuario, tanto en forma independiente (mediante el display, los leds y el reconocimiento de los botones PB1 y PB2), como en forma dependiente (mediante el display y la comunicación serial). Como consecuencia de esto ya existe completa comunicación entre el sistema y cualquier computadora a través del puerto serie y mediante rutinas mostradas en el anexo 2.

Dentro de estas rutinas existen algunas para leer el contenido de memoria de cualquier parte del sistema, rutinas para escribir en cualquier parte de la memoria RAM del sistema, rutinas para verificar los distintos voltajes en los pines de control, etc.

Al existir un prototipo del programa principal únicamente restaría hacer un llamado a la rutina correspondiente en el lugar indicado implementando la función llamada en algún otro lugar del programa.

Se lograron mandar las señales de control, datos y direcciones tanto para lectura como para escritura al socket pero al momento de insertar la EPROM no funcionaban, por lo cual, se sugiere verificar la sincronización adecuada de dichas señales, así como verificar los tiempos y la implementación de los algoritmos de programación.

Debido a los constantes problemas de hardware (chips quemados, en particular PPIs) se aconseja revisar la implementación de la fuente de 5 V. colocándola fuera de la tarjeta para evitar conflictos en la transferencia de información ó en los voltajes de alimentación por exceso de temperatura.

Cabe señalar que para la interfaz de usuario se podría haber usado un número mayor de displays y mas botones, o inclusive un LCD y un teclado controlados por un 8279, en este proyecto no se realizó de esa manera, pero queda abierto para posteriores modificaciones, además de que podría servir muy bien para efectos de un sistema de desarrollo del microcontrolador 8031 realizando las modificaciones pertinentes (programas para la PC, programas para el sistema y algún hardware adicional).

El hecho de que solo se use un socket, se debe a la intención de ahorrar espacio y tiempo ya que todo lo que se lograría con 2 sockets es posible realizarlo con uno solo, aunque para mayor comodidad del usuario sería mejor implementar un segundo socket, pero esto queda para mejoras posteriores.

MANUAL DE USUARIO.

Interfaz con el usuario

En este bloque se localizan los push bottons 1 y 2, así como el display y los leds, de acuerdo al circuito de la figura 5:

El led 1 se ha colocado en consideración a que existiera la conexión al puerto serial de una PC (Computadora Personal), para indicar en cual de los dos modos se está operando; modo local: Copia de EPROM a EPROM sin necesidad de una PC, Modo Remoto: Copia de un archivo en código maquina en una PC a EPROM vía el puerto serial de la PC.

Los otros dos leds Contrapuestos indicarán al usuario el momento en que se están aplicando los voltajes necesarios a la EPROM para evitar que está sea removida en ese momento, el led rojo indicara que se esta leyendo o escribiendo a la EPROM y el led verde indicara el momento en el que se puede quitar o poner la EPROM..

El PB2 (Push Botton 2) y el PB1 servirán como teclado para que el usuario realice la selección del tipo de EPROM, así como el proceso de programación. El PB2 servirá como selector en los menús y submenús y el PB1 como una señal para aceptar la opción elegida por PB2 (ENTER).

El display indicará en cada momento al usuario las diferentes etapas que conforman el proceso de grabar EPROMs, desde seleccionar el modo de trabajo (Dependiente o Independiente). Si se esta en modo independiente, seleccionar el tipo de EPROM con la que se va a trabajar, y el estado en el que se encuentre el proceso de copia. (Selección, lectura, programación, verificación, errores, etc).

FUNCIONAMIENTO:

En base a la explicación de los bloques principales, a continuación se bosqueja el funcionamiento:

Al encender el grabador enviara un mensaje de inicio ("HOLA") al usuario y pondrá en el display un signo de interrogación (?). Esperara que se presione PB2, y estará en modo INDEPENDIENTE (despliega "I"). Al presionar nuevamente PB2 se ira a modo DEPENDIENTE (despliega "d").

Estará oscilando en ese menú mientras no se presione PB1. Si se desea trabajar en modo INDEPENDIENTE se debe presionar PB1 cuando este la "I" en el display de 7 segmentos, en ese momento se entrará al menú del modo INDEPENDIENTE. Si lo que se desea es trabajar en modo DEPENDIENTE, se tendrá que presionar PB1 cuando la "d" este en el display y se entrara al menú del modo DEPENDIENTE. El proceso anterior se describe en el diagrama de flujo 1 que muestra además las acciones que se realizan vía software.

El menú del modo INDEPENDIENTE se describe a continuación:

"G" ----> Indica grabar de EPROM a EPROM.
"C" ----> Indica comparar el contenido de la EPROM con el buffer del grabador.
"c" ----> Es una señal generalizada que indicara salir al menú anterior en todos los menús y submenús.

Para realizar cualquiera de las acciones anteriores se debe de seleccionar con el PB2 y después arrancar el proceso con el PB1. Este proceso lo observamos en el diagrama de flujo 2.

Cada una de las opciones del modo INDEPENDIENTE nos lleva a una rutina. La rutina para GRABAR se muestra en el diagrama de flujo 4.

A continuación de detalla el proceso de GRABAR.

Al iniciar el proceso de programación, el usuario deberá establecer el tipo de EPROM con la que va a trabajar, esto se realiza en base a una tabla de equivalencia la cual se selecciona con el PB2, esta tabla inicia con el tipo 0, y va aumentando hasta agotarla y vuelve a iniciar en forma cíclica, la tabla es la siguiente:

# selección	EPROM	Vpp	# selección	EPROM	Vpp
0	2716	25 V	5	271128	21 V
1	2732	25 V	6	27128A	12.5 V
2	2732A	21 V	7	27256	12.5 V
3	2764	21 V	8	27512	12.5 V
4	2764A	12.5 V			

Después de elegir el tipo de EPROM, se presiona el PB1 para iniciar el proceso, en este punto se pedirá que se cargue la EPROM fuente mediante la aparición de una señal en el display ("F"); a continuación se vuelve a presionar el PB1 y el grabador leerá datos de la EPROM fuente, según el tamaño de la EPROM, durante este proceso estará encendido el led rojo.

Una vez lleno el buffer, aparecerá otra señal en el display ("d") indicando que se cargue la EPROM destino existiendo la posibilidad de elegir la opción de salir presionando PB2 para elegir "c" y después PB1 para salir. Si se elige "d" y ya se cargo la EPROM destino se presiona nuevamente el PB1, el grabador procederá a cargarla con la información del buffer, mientras vuelve a encender el led rojo.

Si existiera un error, el display mostrara alternadamente una "H" y un número 1 el cual indicara que existe un error, se deberá presionar el PB1 para reiniciar el proceso.

Si no existió ningún error, el grabador checará si la EPROM ha sido copiada en su totalidad (solo para EPROMs 27512 que son de 64Kb), y si es así el grabador regresará al inicio del proceso mostrando el tipo de EPROM seleccionado, de lo

contrario se pedirá al usuario que inserte la EPROM fuente nuevamente y todo el proceso anterior se repetirá hasta copiar completamente el contenido de la EPROM.

El hecho de que solo se use un socket, se debe a la intención de ahorrar espacio y tiempo ya que todo lo que se lograría con 2 sockets es posible realizarlo con uno solo, aunque para mayor comodidad del usuario seria mejor implementar un segundo socket, pero esto queda para mejoras posteriores.

Si se eligió en el menú principal la opción de trabajar en modo dependiente, se mostrara una "E" que indica "E"spera un caracter del puerto serial. En este momento se deberá correr el programa "ITFC.EXE" para inrtefazar la PC con el grabador.

Este programa muestra un menú con las opciones de:

ARCHIVO:	Abrir	*
	Guardar como	
	Cargar al buffer	*
EPROM:	Programar	
	Verificar copia	
	Cargar a buffer	
	Imprimir	
BUFFER:	Editar	*
	Imprimir	
AYUDA	Acerca de	
	Manual	

La opción Abrir, del menú ARCHIVO, abre un archivo con información en HEX, y lo alista para Cargarlo al buffer del grabador.

La opción Editar del menú BUFFER carga a la RAM de la PC 80 bytes de código HEX a partir de una dirección especificada por el usuario y los despliega en la pantalla.

Las opciones marcadas con * son las que están implementadas en el programa ITFC.PAS.

BIBLIOGRAFÍA.

- Intel. MICROCONTROLLER HANDBOOK. Familia MCS-51,
  Capítulo 7.- Architecture MCS-51
  Capitulo 8.- MCS-51 Programmer's Guide and Instruction Set.
  Capitulo 9.- Data Sheets.
  Capitulo 10.- MCS-51 Aplication Notes.

- Intel MEMORY.
  Capitulo 3.- Dinamic and Static RAMs (Random Access Memories)
  51256 (pag. 3-80), 5164 (pag. 3-37).
  Capitulo 4.- EPROMs (Erasable Programmable Read Only Memories)
  2716 (pag. 4-1), 2732 (pag. 4-9). 2764 (pag. 4.18),
  27128 (pag. 4-42), 27256 (pag. 463), 27512 (pag. 4-111)
  Algoritmo de programación Quick-Pulse (pag. 4-397)

- Intel. DISPOSITIVOS PERIFÉRICOS.
  PPI (Programmable Peripherical Interface) 8255. Pags 3-100 a 3-119.

- SGS DATABOOK, LOW POWER SCHOTTKY TTL ICs
  C.I. : 74373, 74245, 74138, 7400, 7404, 7408. 7407. 74LS151.

- William G. Houghton MASTERING DIGITAL DEVICE CONTROL.
  Capitulo 1.- The Intel 8051 Family.
  Capitulo 2.- External Program Memory Expansion
  Capitulo 3.- External Data Memory Expansion
  Capitulo 4.- Expanding I/O
  Capitulo 8.- Adding An RS-232 Port

- Motorola. SEMICONDUCTOR TECHNICAL DATA.
  MUA78S40 pag. 3-330
  LM317 pag. 3-21
  LM337 pag. 3-43

- Intronics, inc. INTRODUCING TO EPROM PROGRAMER.
  Pag. 1 - 11.

ANEXO 1

JUSTIFICACIÓN
Y
HOJAS TÉCNICAS
DE
CIRCUITOS INTEGRADOS

8031:

- microcontrolador de 8 bits.
- 4 puertos de 8 bits cada uno.
- Memoria RAM interna de 128 x 8 bits.
- 2 timers de 16 bits.
- Interrupciones.
- Tecnología HMOS.

El uso de este chip se basa en la existencia y precio que de el se obtuvo en el mercado. así como de contar con las terminales necesarias para manejar información de 8 bits, y direcciones de 16 bits. que son características del grabador.

74245:

- Buffer bidireccional octal de tres estados.
- Terminal para habilitar salidas.
- Control para transmisión y recepción.
- Canal bidireccional de 8 bits.
- Estado de alta impedancia.

Este circuito, debido a sus 8 bits y a sus tres estado, permite el intercambio de información, en este caso del microcontrolador con el resto del grabador.

74373:

- Latch octal con salida en tercer estado.
- Control de entradas al latch.
- Control de salidas del latch.
- Canal de 8 bits para datos de entrada
- Canal de 8 bits para datos de salida (en tercer estado).

Para este circuito se toma en cuenta su capacidad para el manejo de 8 bits y su estado de alta impedancia.

74138:

- Decoficicador y demultiplexor de tres a ocho.
- Tres entradas de control para direcciones.
- Tres entradas de habilitación.
- Ocho salidas posibles.

El uso de este circuito se basa en las necesidades de poder controlar hasta ocho dispositivos con solo tres líneas para habilitarlos.

8255:

- Interface periférica programable.
- 3 buses de 8 bits.
- Control de lectura.
- Control de Escritura.
- Control de selección.
- Buses con estado de alta impedancia.
- Línea de reset.

- 2 líneas de selección de bus.

Este circuito integrado nos permite realizar la comunicación entre la parte que maneja el usuario, y el socket del grabador con el microcontrolador.

2764:

- Capacidad para 8Kb x 8bits
- CHMOS compatible con microprocesadores y microcontroladores.
- Latch de direcciones integrados.
- Tamaño universal de 28 pines con dos líneas de control.
- Bajo consumo de potencia (100 microA máximo).
- Características de inmunidad al ruido.
- Alta velocidad de respuesta.

Esta memoria tipo EPROM tiene las características necesarias para adaptarla a nuestro sistema y almacenara el programa principal (BIOS) del sistema grabador..

6264:

- Capacidad para 8Kb x 8bits
- Operación estática.
- Tiempos iguales de acceso para lectura y escritura.
- 5 volts de alimentación.
- Compatible con TTL.
- Datos comunes de entrada y salida.

Esta memoria tipo RAM tiene la usamos únicamente con propósitos de desarrollo del sistema, es decir, sirve como almacén temporal del programa principal para realizar pruebas. ya terminado el sistema no será necesario su uso.

62256:

- Capacidad para 32Kb x 8bits
- Operación estática.
- Tiempos iguales de acceso para lectura y escritura.
- 5 volts de alimentación.
- Compatible con TTL.
- Datos comunes de entrada y salida.

Esta memoria tipo RAM tiene la finalidad de servir como buffer del sistema grabador, es decir, será donde se almacenen los datos del programa a grabarse en la EPROM.

LM337:

- Corriente de salida mayor a 1.5 A.
- Salida ajustable entre -1.2V y -37V.
- Protección térmica interna.
- Corriente constante con la temperatura.
- Operación flotante para aplicaciones de alto voltaje.

Este regulador de voltaje lo usamos para proporcionar a los transistores que controlan los voltajes un voltaje de -1.4V.

LM317:

- Corriente de salida mayor a 1.5 A.
- Salida ajustable entre -1.2V y -37V.
- Protección térmica interna.
- Corriente constante con la temperatura.
- Operación flotante para aplicaciones de alto voltaje.

Este regulador de voltaje lo usamos para proporcionar a los pines del socket los voltajes de programación adecuados controlados por los transistores.

MUA78S40:

- Corriente de salida de 1.5 A sin transistor de salida.
- Salida ajustable entre 1.5V y 40V.
- Línea de 80dB y protección de carga.
- Soporta desde 2.5V hasta 40V de entrada.
- Alta ganancia.

Este convertidor de DC a DC lo usamos para generar los 30 volts que necesitan en la entrada de los reguladores LM317.

Además de los circuitos integrados mencionados anteriormente, se utilizaran algunas compuertas lógicas tales como inversores, NAND, OR, etc. cuya elección dependerá del uso inmediato que se proyecte, así como algunos otros componentes de acuerdo a la hoja de especificaciones para el buen funcionamiento de los circuitos.

intel

PRELIMINARY

MCS®-51
8-BIT CONTROL-ORIENTED MICROCOMPUTERS

8031/8051
8031AH/8051AH
8032AH/8052AH
8751H/8751H-12/8751H-88

- **High Performance HMOS Process**
- **Internal Timers/Event Counters**
- **2-Level Interrupt Priority Structure**
- **32 I/O Lines (Four 8-Bit Ports)**
- **64K Program Memory Space**
- **Boolean Processor**
- **Bit-Addressable RAM**
- **Programmable Full Duplex Serial Channel**
- **111 Instructions (64 Single-Cycle)**
- **64K Data Memory Space**

- **Security Feature Protects EPROM Parts Against Software Piracy**

The MCS®-51 products are optimized for control applications. Byte-processing and numerical operations on small data structures are facilitated by a variety of fast addressing modes for accessing the internal RAM. The instruction set provides a convenient menu of 8-bit arithmetic instructions, including multiply and divide in structions. Extensive on-chip support is provided for one-bit variables as a separate data type, allowing direct bit manipulation and testing in control and logic systems that require Boolean processing.

	Internal Memory			
Device	Program	Data	Timers/ Event Counters	Interrupts
8052AH	8K × 8 ROM	256 × 8 RAM	3 × 16-Bit	6
8051AH	4K × 8 ROM	128 × 8 RAM	2 × 16-Bit	5
8051	4K × 8 ROM	128 × 8 RAM	2 × 16-Bit	5
8032AH	none	256 × 8 RAM	3 × 16-Bit	6
8031AH	none	128 × 8 RAM	2 × 16-Bit	5
8031	none	128 × 8 RAM	2 × 16-Bit	5
8751H	4K × 8 EPROM	128 × 8 RAM	2 × 16-Bit	5
8751H-12	4K × 8 EPROM	128 × 8 RAM	2 × 16-Bit	5
8751H-88	4K × 8 EPROM	128 × 8 RAM	2 × 16-Bit	5

The 8751H is an EPROM version of the 8051AH; that is, the on-chip Program Memory can be electrically programmed, and can be erased by exposure to ultraviolet light. It is fully compatible with its predecessor, the 8751-8, but incorporates two new features: a Program Memory Security bit that can be used to protect the EPROM against unauthorized read-out, and a programmable baud rate modification bit (SMOD). SMOD is not present in the 8751H-12 or the 8751H-88. The 8751H-88 also only operates up to 8 MHz.

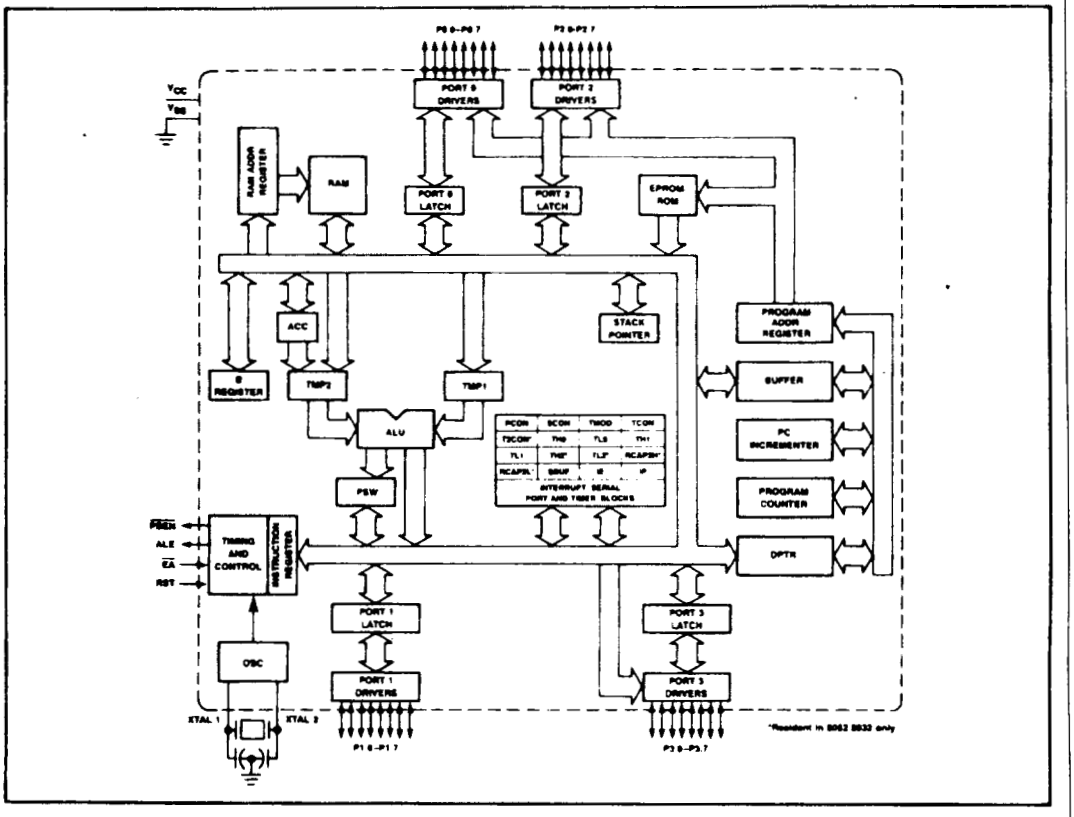


Figure 1. MCS®-51 Block Diagram

PIN DESCRIPTIONS

VCC

Supply voltage.

VSS

Circuit ground.

Port 0

Port 0 is an 8-bit open drain bidirectional I/O port. As an output port each pin can sink 8 LS TTL inputs. Port 0 pins that have 1s written to them float, and in that state can be used as high-impedance inputs.

Port 0 is also the multiplexed low-order address and data bus during accesses to external Program and Data Memory. In this application it uses strong internal pullups when emitting 1s, and can source and sink 8 LS TTL inputs.

Port 0 also receives the code bytes during programming of the EPROM parts, and outputs the code bytes during program verification of the ROM and EPROM parts. External pullups are required during program verification.

Port 1

Port 1 is an 8-bit bidirectional I/O port with internal pullups. The Port 1 output buffers can sink/source 4 LS TTL inputs. Port 1 pins that have 1s written to them are pulled high by the internal pullups, and in that state can be used as inputs. As inputs, Port 1 pins that are externally being pulled low will source current (IIL, on the data sheet) because of the internal pullups.

Port 1 also receives the low-order address bytes during programming of the EPROM parts and during program verification of the ROM and EPROM parts.

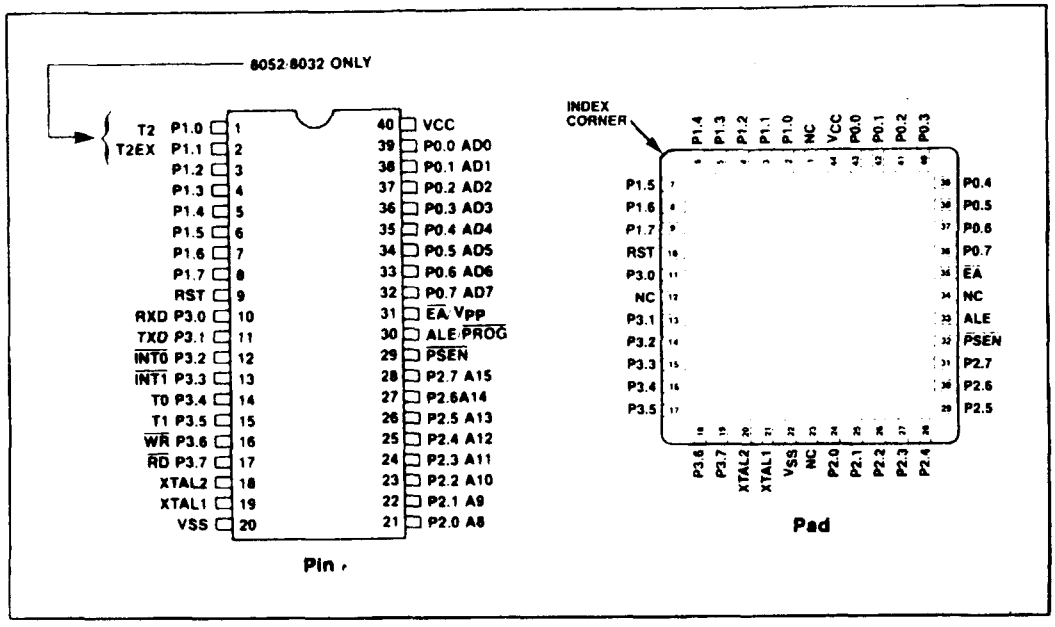


Figure 2. MCS®-51 Connections

In the 8032AH and 8052AH, Port 1 pins P1.0 and P1.1 also serve the T2 and T2EX functions, respectively.

Port 2

Port 2 is an 8-bit bidirectional I/O port with internal pullups. The Port 2 output buffers can sink/source 4 LS TTL inputs. Port 2 pins that have 1s written to them are pulled high by the internal pullups, and in that state can be used as inputs. As inputs, Port 2 pins that are externally being pulled low will source current (IIL, on the data sheet) because of the internal pullups.

Port 2 emits the high-order address byte during fetches from external Program Memory and during accesses to external Data Memory that use 16-bit addresses (MOVX @DPTR). In this application it uses strong internal pullups when emitting 1s. During accesses to external Data Memory that use 8-bit addresses (MOVX @Ri), Port 2 emits the contents of the P2 Special Function Register.

Port 2 also receives the high-order address bits during programming of the EPROM parts and during program verification of the ROM and EPROM parts.

Port 3

Port 3 is an 8-bit bidirectional I/O port with internal pullups. The Port 3 output buffers can sink/source 4 LS TTL inputs. Port 3 pins that have 1s written to them are pulled high by the internal pullups, and in that state can be used as inputs. As inputs, Port 3 pins that are externally being pulled low will source current (IIL, on the data sheet) because of the pullups.

Port 3 also serves the functions of various special features of the MCS-51 Family, as listed below:

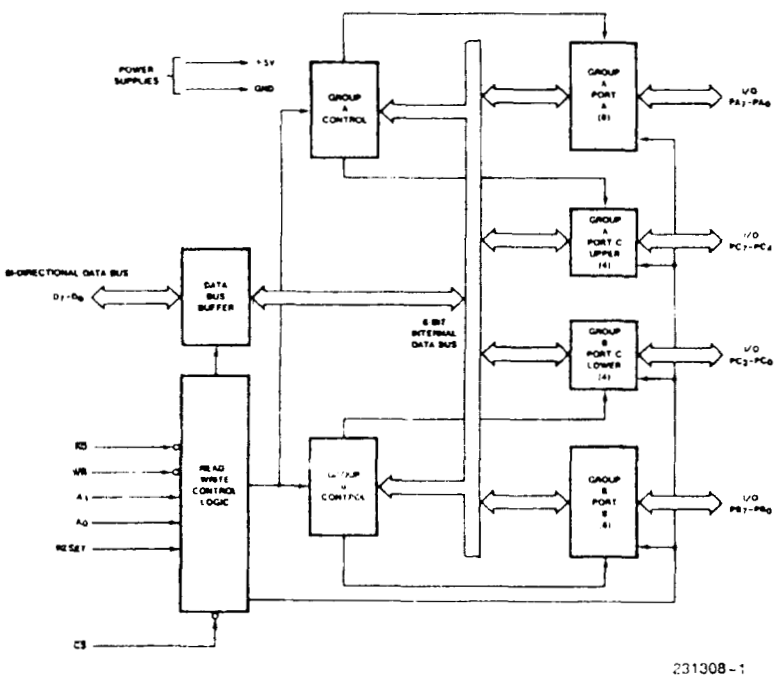
Port Pin	Alternative Function
P3.0	RXD (serial input port)
P3.1	TXD (serial output port)
P3.2	$\overline{INT0}$ (external interrupt 0)
P3.3	$\overline{INT1}$ (external interrupt 1)
P3.4	T0 (Timer 0 external input)
P3.5	T1 (Timer 1 external input)
P3.6	$\overline{WR}$ (external data memory write strobe)
P3.7	$\overline{RD}$ (external data memory read strobe)

8255A/8255A-5
PROGRAMMABLE PERIPHERAL INTERFACE

- MCS-85™ Compatible 8255A-5
- 24 Programmable I/O Pins
- Completely TTL Compatible
- Fully Compatible with Intel Microprocessor Families
- Improved Timing Characteristics
- Direct Bit Set/Reset Capability Easing Control Application Interface
- Reduces System Package Count
- Improved DC Driving Capability
- Available in EXPRESS
  - Standard Temperature Range
  - Extended Temperature Range
- 40 Pin DIP Package or 44 Lead PLCC

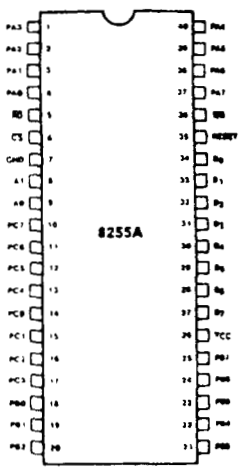
(See Intel Packaging: Order Number 231369)

The Intel 8255A is a general purpose programmable I/O device designed for use with Intel microprocessors. It has 24 I/O pins which may be individually programmed in 2 groups of 12 and used in 3 major modes of operation. In the first mode (MODE 0), each group of 12 I/O pins may be programmed in sets of 4 to be input or output. In MODE 1, the second mode, each group may be programmed to have 8 lines of input or output. Of the remaining 4 pins, 3 are used for handshaking and interrupt control signals. The third mode of operation (MODE 2) is a bidirectional bus mode which uses 8 lines for a bidirectional bus, and 5 lines, borrowing one from the other group, for handshaking.



231308-1

Figure 1. 8255A Block Diagram



231308-2

Figure 2. Pin Configuration

November 1987
Order Number: 231308-002

8255A FUNCTIONAL DESCRIPTION

General

The 8255A is a programmable peripheral interface (PPI) device designed for use in Intel microcomputer systems. Its function is that of a general purpose I/O component to interface peripheral equipment to the microcomputer system bus. The functional configuration of the 8255A is programmed by the system software so that normally no external logic is necessary to interface peripheral devices or structures.

Data Bus Buffer

This 3-state bidirectional 8-bit buffer is used to interface the 8255A to the system data bus. Data is transmitted or received by the buffer upon execution of input or output instructions by the CPU. Control words and status information are also transferred through the data bus buffer.

Read/Write and Control Logic

The function of this block is to manage all of the internal and external transfers of both Data and Control or Status words. It accepts inputs from the CPU Address and Control busses and in turn, issues commands to both of the Control Groups.

($\overline{CS}$)

Chip Select. A "low" on this input pin enables the communication between the 8255A and the CPU.

($\overline{RD}$)

Read. A "low" on this input pin enables the 8255A to send the data or status information to the CPU on the data bus. In essence, it allows the CPU to "read from" the 8255A.

($\overline{WR}$)

Write. A "low" on this input pin enables the CPU to write data or control words into the 8255A.

(A_0 and A_1)

Port Select 0 and Port Select 1. These input signals, in conjunction with the RD and WR inputs, control the selection of one of the three ports or the control word registers. They are normally connected to the least significant bits of the address bus (A_0 and A_1).

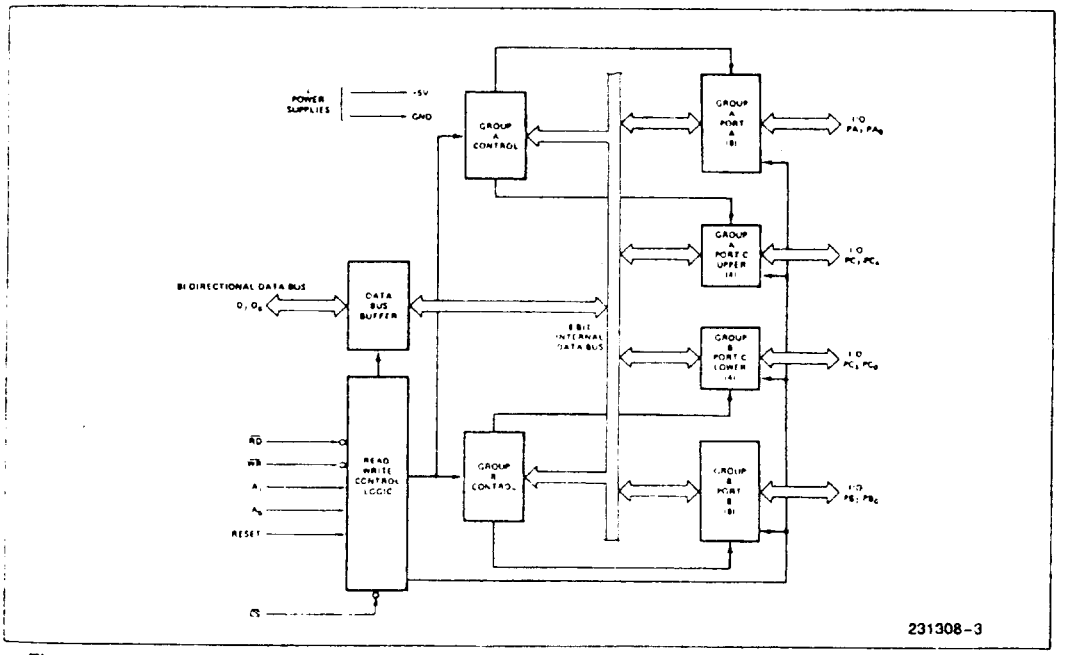


Figure 3. 8255A Block Diagram Showing Data Bus Buffer and Read/Write Control Logic Functions

8255A BASIC OPERATION

A_1	A_0	$\overline{RD}$	$\overline{WR}$	$\overline{CS}$	Input Operation (READ)
0	0	0	1	0	Port A → Data Bus
0	1	0	1	0	Port B → Data Bus
1	0	0	1	0	Port C → Data Bus
					Output Operation (WRITE)
0	0	1	0	0	Data Bus → Port A
0	1	1	0	0	Data Bus → Port B
1	0	1	0	0	Data Bus → Port C
1	1	1	0	0	Data Bus → Control
					Disable Function
X	X	X	X	1	Data Bus → 3-State
1	1	0	1	0	Illegal Condition
X	X	1	1	0	Data Bus → 3-State

(RESET)

Reset. A "high" on this input clears the control register and all ports (A, B, C) are set to the input mode.

Group A and Group B Controls

The functional configuration of each port is programmed by the systems software. In essence, the CPU "outputs" a control word to the 8255A. The control word contains information such as "mode", "bit set", "bit reset", etc., that initializes the functional configuration of the 8255A.

Each of the Control blocks (Group A and Group B) accepts "commands" from the Read/Write Control Logic, receives "control words" from the internal data bus and issues the proper commands to its associated ports.

Control Group A—Port A and Port C upper (C7–C4)

Control Group B—Port B and Port C lower (C3–C0)

The Control Word Register can **Only** be written into. No Read operation of the Control Word Register is allowed.

Ports A, B, and C

The 8255A contains three 8-bit ports (A, B, and C). All can be configured in a wide variety of functional characteristics by the system software but each has its own special features or "personality" to further enhance the power and flexibility of the 8255A.

Port A. One 8-bit data output latch/buffer and one 8-bit data input latch.

Port B. One 8-bit data input/output latch/buffer and one 8-bit data input buffer.

Port C. One 8-bit data output latch/buffer and one 8-bit data input buffer (no latch for input). This port can be divided into two 4-bit ports under the mode control. Each 4-bit port contains a 4-bit latch and it can be used for the control signal outputs and status signal inputs in conjunction with ports A and B.

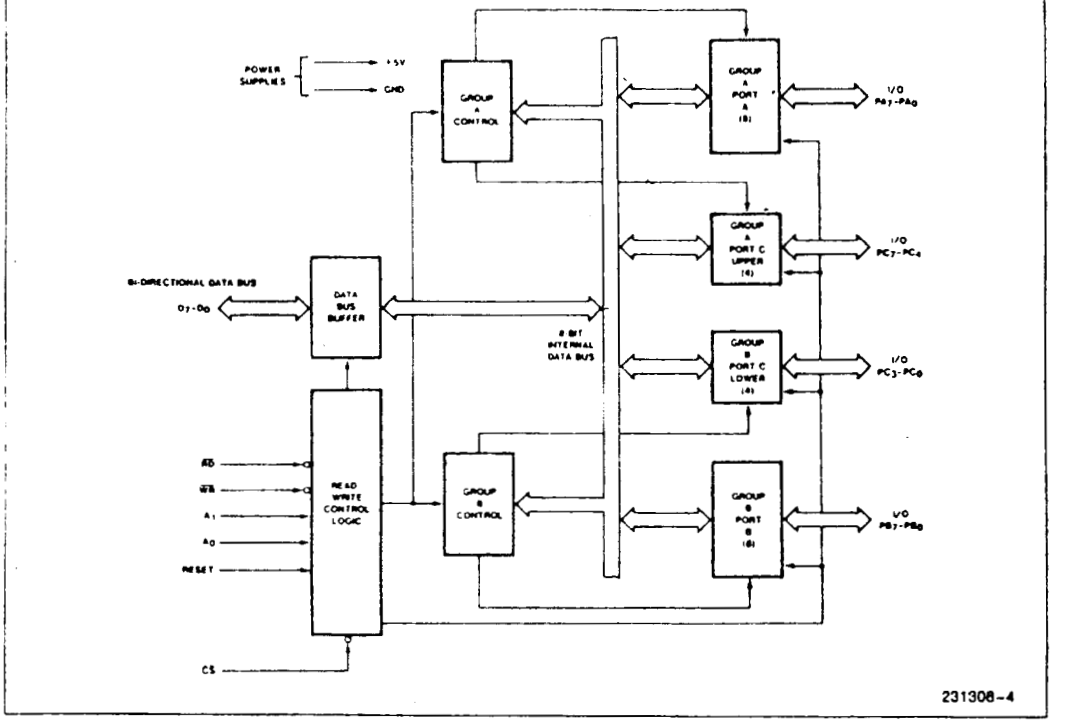
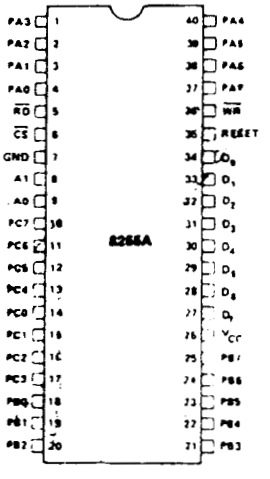


Figure 4. 8225A Block Diagram Showing Group A and Group B Control Functions

Pin Configuration



231308-5

Pin Names

D_7-D_0	Data Bus (Bi-Directional)
RESET	Reset Input
$\overline{CS}$	Chip Select
$\overline{RD}$	Read Input
$\overline{WR}$	Write Input
A0, A1	Port Address
PA7–PA0	Port A (BIT)
PB7–PB0	Port B (BIT)
PC7–PC0	Port C (BIT)
V_{CC}	+5 Volts
GND	0 Volts

8255A OPERATIONAL DESCRIPTION

Mode Selection

There are three basic modes of operation that can be selected by the system software:

Mode 0—Basic Input/Output

Mode 1—Strobed Input/Output

Mode 2—Bi-Directional Bus

When the reset input goes "high" all ports will be set to the input mode (i.e., all 24 lines will be in the high impedance state). After the reset is removed the 8255A can remain in the input mode with no additional initialization required. During the execution of the system program any of the other modes may be selected using a single output instruction. This allows a single 8255A to service a variety of peripheral devices with a simple software maintenance routine.

The modes for Port A and Port B can be separately defined, while Port C is divided into two portions as required by the Port A and Port B definitions. All of the output registers, including the status flip-flops, will be reset whenever the mode is changed. Modes may be combined so that their functional definition can be "tailored" to almost any I/O structure. For instance; Group B can be programmed in Mode 0 to monitor simple switch closings or display computational results, Group A could be programmed in Mode 1 to monitor a keyboard or tape reader on an interrupt-driven basis.

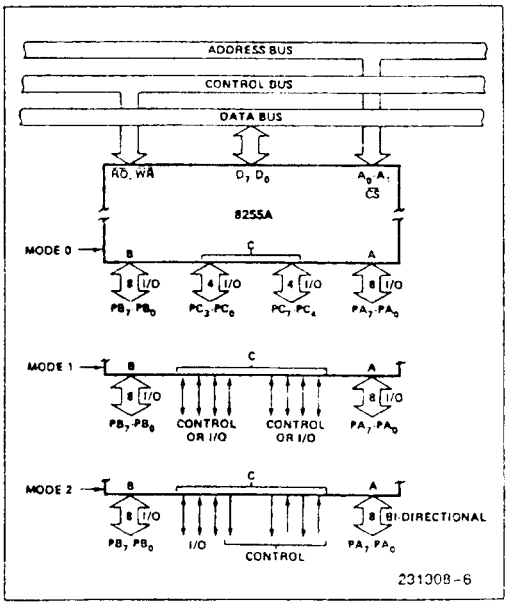


Figure 5. Basic Mode Definitions and Bus Interface

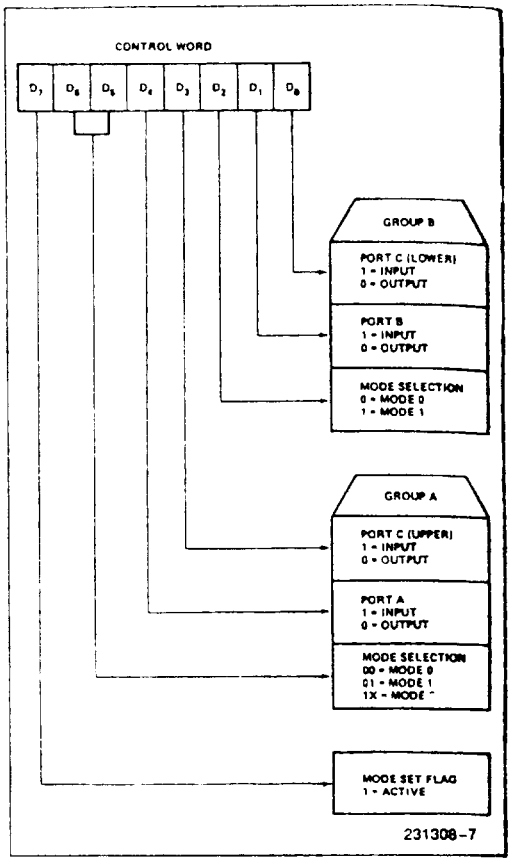


Figure 6. Mode Definition Format

The mode definitions and possible mode combinations may seem confusing at first but after a cursory review of the complete device operation a simple, logical I/O approach will surface. The design of the 8255A has taken into account things such as efficient PC board layout, control signal definition vs PC layout and complete functional flexibility to support almost any peripheral device with no external logic. Such design represents the maximum use of the available pins.

Single Bit Set/Reset Feature

Any of the eight bits of Port C can be Set or Reset using a single OUTput instruction. This feature reduces software requirements in Control-based applications.

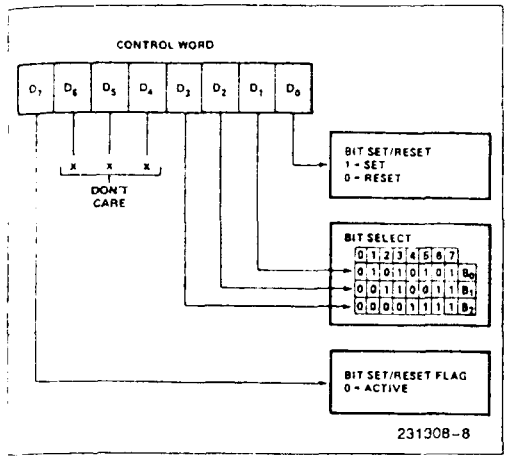


Figure 7. Bit Set/Reset Format

When Port C is being used as status/control for Port A or B, these bits can be set or reset by using the Bit Set/Reset operation just as if they were data output ports.

Interrupt Control Functions

When the 8255A is programmed to operate in mode 1 or mode 2, control signals are provided that can be used as interrupt request inputs to the CPU. The interrupt request signals, generated from port C, can be inhibited or enabled by setting or resetting the associated INTE flip-flop, using the bit set/reset function of port C.

This function allows the Programmer to disallow or allow a specific I/O device to interrupt the CPU without affecting any other device in the interrupt structure.

INTE flip-flop definition:

(BIT-SET)—INTE is set—Interrupt enable

(BIT-RESET)—INTE is RESET—Interrupt disable

NOTE:
All Mask flip-flops are automatically reset during mode selection and device Reset.

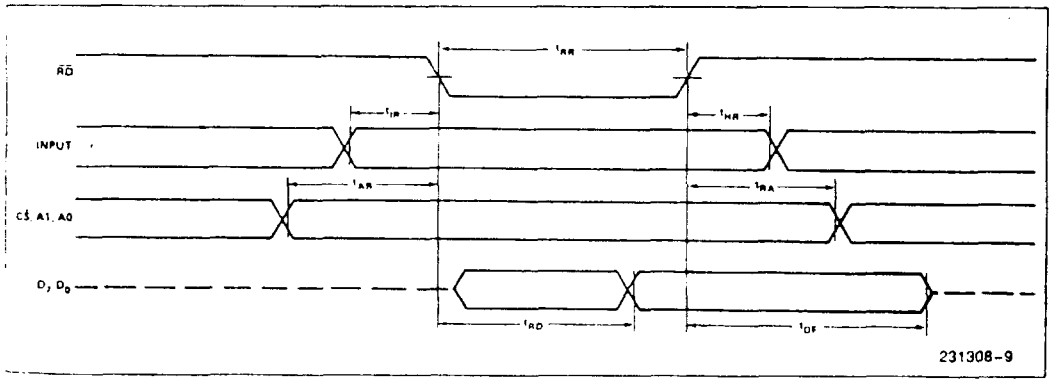
Operating Modes

MODE 0 (Basic Input/Output). This functional configuration provides simple input and output operations for each of the three ports. No "handshaking" is required, data is simply written to or read from a specified port.

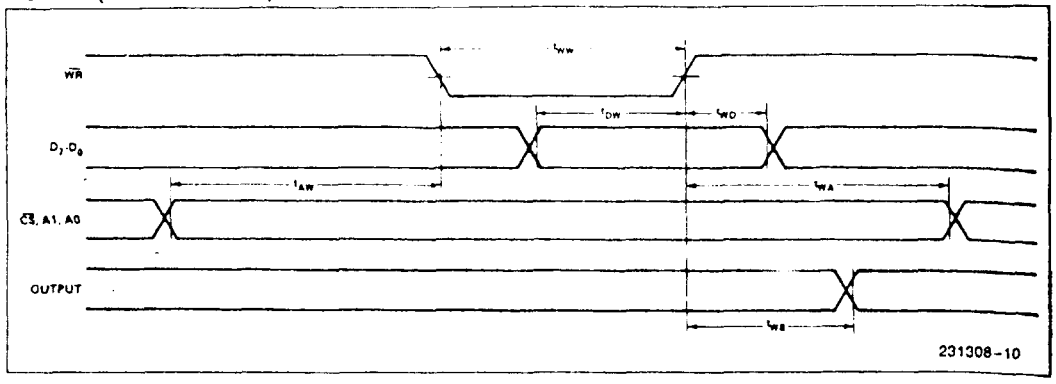
Mode 0 Basic Functional Definitions:

- Two 8-bit ports and two 4-bit ports.
- Any port can be input or output.
- Outputs are latched.
- Inputs are not latched.
- 16 different Input/Output configurations are possible in this Mode.

MODE 0 (BASIC INPUT)



MODE 0 (BASIC OUTPUT)



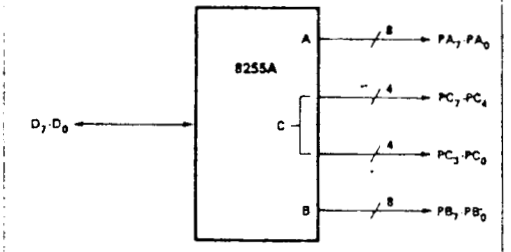
MODE 0 PORT DEFINITION

A		B		Group A			Group B	
D_4	D_3	D_1	D_0	Port A	Port C (Upper)	#	Port B	Port C (Lower)
0	0	0	0	OUTPUT	OUTPUT	0	OUTPUT	OUTPUT
0	0	0	1	OUTPUT	OUTPUT	1	OUTPUT	INPUT
0	0	1	0	OUTPUT	OUTPUT	2	INPUT	OUTPUT
0	0	1	1	OUTPUT	OUTPUT	3	INPUT	INPUT
0	1	0	0	OUTPUT	INPUT	4	OUTPUT	OUTPUT
0	1	0	1	OUTPUT	INPUT	5	OUTPUT	INPUT
0	1	1	0	OUTPUT	INPUT	6	INPUT	OUTPUT
0	1	1	1	OUTPUT	INPUT	7	INPUT	INPUT
1	0	0	0	INPUT	OUTPUT	8	OUTPUT	OUTPUT
1	0	0	1	INPUT	OUTPUT	9	OUTPUT	INPUT
1	0	1	0	INPUT	OUTPUT	10	INPUT	OUTPUT
1	0	1	1	INPUT	OUTPUT	11	INPUT	INPUT
1	1	0	0	INPUT	INPUT	12	OUTPUT	OUTPUT
1	1	0	1	INPUT	INPUT	13	OUTPUT	INPUT
1	1	1	0	INPUT	INPUT	14	INPUT	OUTPUT
1	1	1	1	INPUT	INPUT	15	INPUT	INPUT

MODE CONFIGURATIONS

CONTROL WORD #0

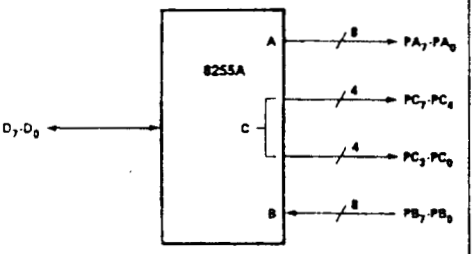
D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	0	0	0	0	0



231308-11

CONTROL WORD #2

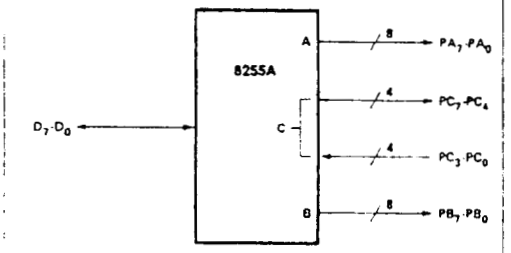
D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	0	0	0	1	0



231308-12

CONTROL WORD #1

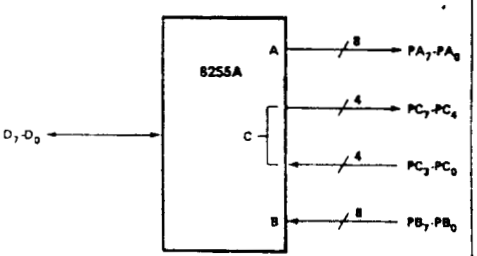
D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	0	0	0	0	1



231308-13

CONTROL WORD #3

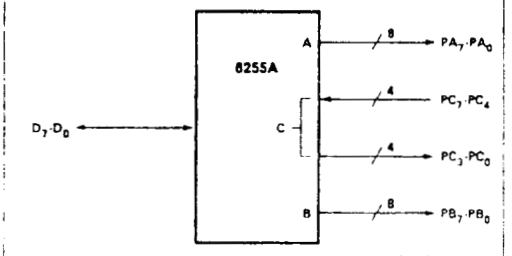
D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	0	0	0	1	1



231308-14

CONTROL WORD #4

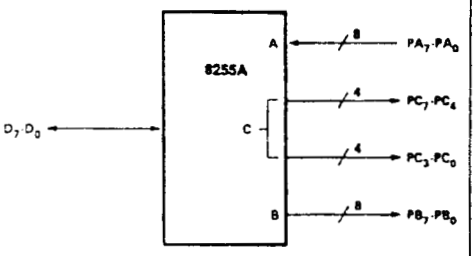
D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	0	1	0	0	0



231308-15

CONTROL WORD #8

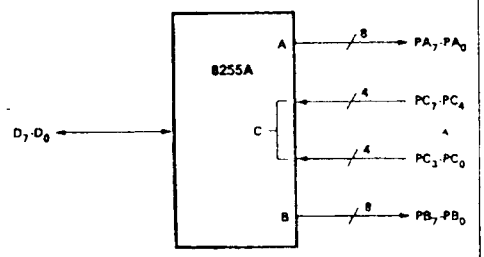
D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	1	0	0	0	0



231308-16

CONTROL WORD #5

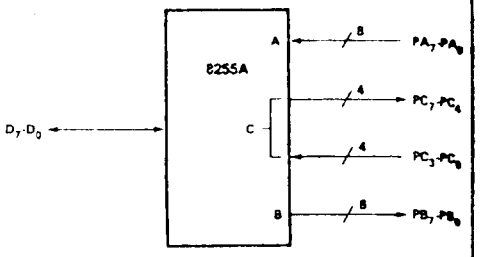
D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	0	1	0	0	1



231308-17

CONTROL WORD #9

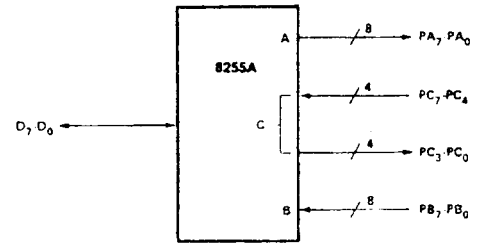
D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	1	0	0	0	1



231308-18

CONTROL WORD #6

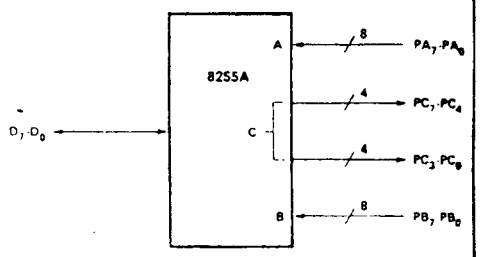
D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	0	1	0	1	0



231308-19

CONTROL WORD #10

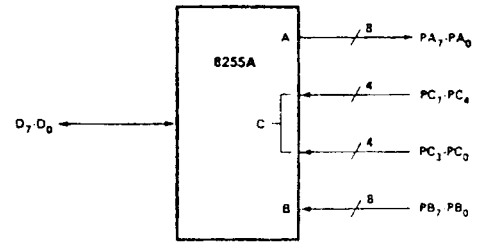
D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	1	0	0	1	0



231308-20

CONTROL WORD #7

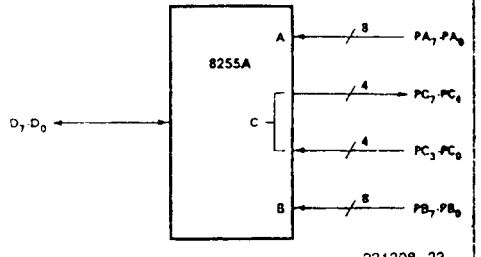
D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	0	1	0	1	1



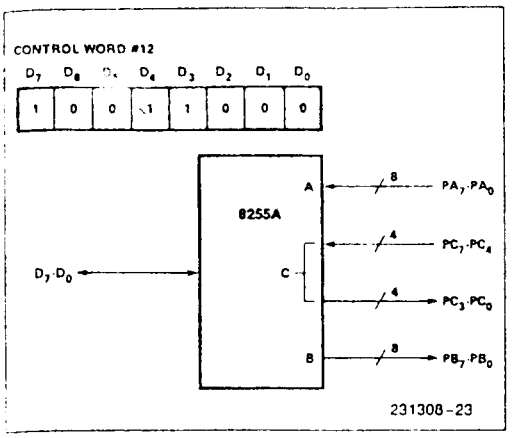
231308-21

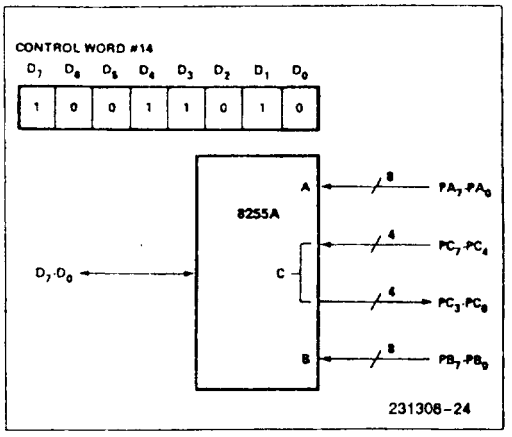
CONTROL WORD #11

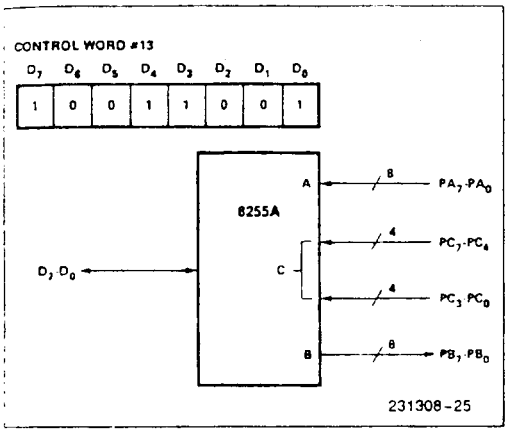
D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	1	0	0	1	1

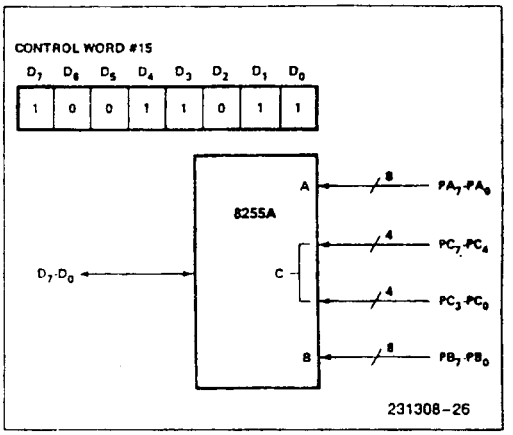


231308-22









Operating Modes

MODE 1 (Strobed Input/Output). This functional configuration provides a means for transferring I/O data to or from a specified port in conjunction with strobes or "handshaking" signals. In mode 1, port A and port B use the lines on port C to generate or accept these "handshaking" signals.

Mode 1 Basic Functional Definitions:

- Two Groups (Group A and Group B)
- Each group contains one 8-bit data port and one 4-bit control/data port.
- The 8-bit data port can be either input or output. Both inputs and outputs are latched.
- The 4-bit port is used for control and status of the 8-bit data port.

Input Control Signal Definition

$\overline{STB}$ (Strobe Input). A "low" on this input loads data into the input latch.

IBF (Input Buffer Full F/F)

A "high" on this output indicates that the data has been loaded into the input latch; in essence, an acknowledgement. IBF is set by STB input being low and is reset by the rising edge of the RD input.

INTR (Interrupt Request)

A "high" on this output can be used to interrupt the CPU when an input device is requesting service. INTR is set by the $\overline{STB}$ is a "one", IBF is a "one" and INTE is a "one". It is reset by the falling edge of $\overline{RD}$. This procedure allows an input device to request service from the CPU by simply strobing its data into the port.

5164S/L
8K x 8-BIT CMOS STATIC RAM

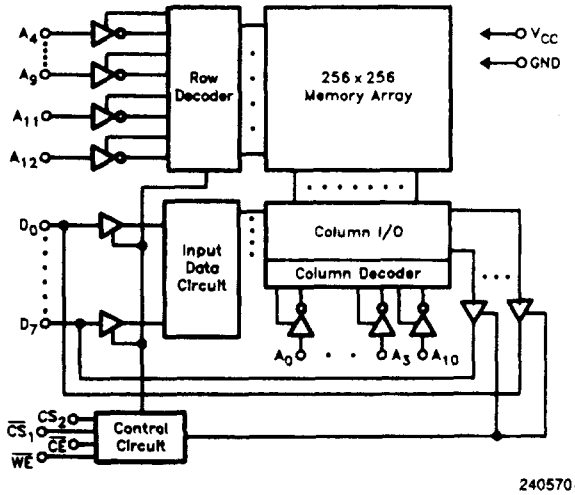
	5164S-10	5164S-12	Units
Address Access Time (t_{AA})	100	120	ns
Chip Select Access Time (t_{ACS})	100	120	ns
Output Enable Access Time (t_{OE})	55	60	ns

- **Static Operation — No Clock/Refresh Required**
- **Equal Access and Cycle Times — Simplifies System Design**
- **Single +5V Supply**
- **Power Down Mode**
- **TTL Compatible**
- **Common Data Input and Output**
- **High Reliability 28-Pin 600 Mil PDIP Package**

The 5164S is a 8192-word by 8-bit CMOS static RAM fabricated using CMOS Silicon Gate process.

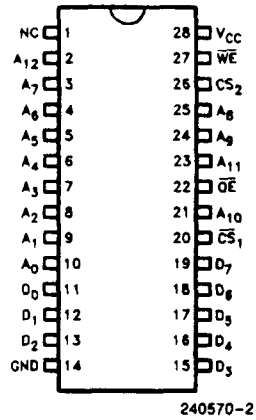
The 5164S is placed in a standby or reduced power consumption mode by asserting either CS input ($\overline{CS_1}$, CS_2) false. When in standby mode, the device is deselected and the outputs are in a high impedance state, independent of the $\overline{WE}$ input. When device is deselected, standby current is reduced to 100 μA (max). The device will remain in standby mode until both pins are asserted true again. The device has a data retention mode that guarantees that data will remain valid at minimum V_{CC} of 2.0V.

Block Diagram



240570-1

Pin Connections



240570-2

Pin Names

A_0–A_{12}	Address Input
D_0–D_7	Data Input/Output
$\overline{CS_1}$	Chip Select One
CS_2	Chip Select Two
$\overline{WE}$	Write Enable
$\overline{OE}$	Output Enable
V_{CC}	Power
GND	Ground

Device Operation

The 5164S has three control inputs: Two Chip Selects ($\overline{CS}_1$, CS_2) and Write Enable ($\overline{WE}$). $\overline{WE}$ is the data control pin and should be used to gate data at the I/O pins. A write cycle starts at the lowest transition of $\overline{CS}_1$, low $\overline{WE}$ or high CS_2 and ends at the earliest transitiion of $\overline{CS}_1$, high WE or low CS_2. Out Enable (OE) is used for precise control of the outputs.

The availability of active high and active low chip enable pins provides more system design flexibility than single chip enable devices.

Table 1. Mode Selection Truth Table

$\overline{CS}_1$	CS_2	$\overline{WE}$	$\overline{OE}$	Mode	I/O	Power
H	X	X	X	Standby	High Z	Standby
X	L	X	X	Standby	High Z	Standby
L	H	L	X	Write	D_{IN}	Active
L	H	H	L	Read	D_{OUT}	Active
L	H	H	H	Read	High Z	Active

ABSOLUTE MAXIMUM RATINGS

Voltage on Any Pin
Relative to Ground (V_{IN}, V_{OUT}) ..... −0.3V to 7V
Storage Temperature (T_{STG}) .... −55°C to +150°C
Power Dissipation (P_D) 1.0W
DC Continuous Output Current (I_{OS}) ..... 50 mA

**Notice: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.*

RECOMMENDED OPERATING CONDITIONS

Voltage referenced to V_{SS}, T_A = 0°C to 70°C

Symbol	Parameter	Min	Typ	Max	Units
V_{CC}	Supply Voltage	4.5	5.0	5.5	V
V_{SS}	Ground	0	0	0	V
V_{IH}	Input High Voltage	2.2	—	V_{CC} + 0.3	V
V_{IL}	Input Low Voltage	−0.3	—	0.8	V

NOTE:
1. During transitions, the inputs may undershoot to −3.5V for periods less than 20 ns.

CAPACITANCE T_A = 25°C, f = 1.0 MHz

Symbol	Parameter	Min	Max	Units
C_{IN1}	Input Capacitance (V_{IN} = 0V)	—	6	pF
C_{OUT}	Output Capacitance (V_{OUT} = 0V)	—	8	pF

NOTE:
This parameter is sampled and not 100% tested.

51256S/L
32K x 8-BIT CMOS STATIC RAM

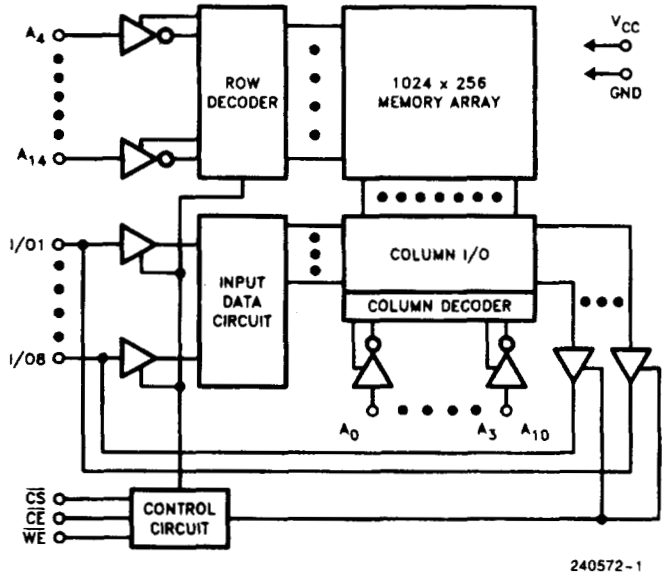
	51256S-10	51256S-12	Unit
Address Access Time (t_{AA})	100	120	ns
Chip Select Access Time (t_{ACS})	100	120	ns
Output Enable Access Time (t_{OE})	50	60	ns

- **Static Operation — No Clock/Refresh Required**
- **Equal Access and Cycle Times — Simplifies System Design**
- **Single +5V Supply**
- **Power Down Mode**
- **TTL Compatible**
- **Common Data Input and Output**
- **High Reliability 28-Pin 600 Mil PDIP Package**

The 51256S is a 32768-word by 8-bit CMOS static RAM fabricated using CMOS Silicon Gate process.

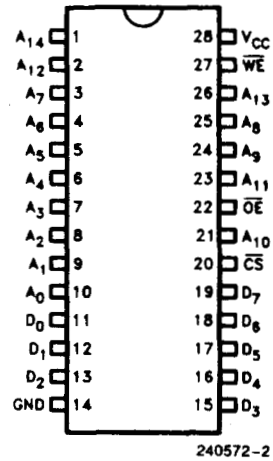
When the Chip Select is brought high, the device assumes a standby mode in which the standby current is reduced to 100 μA (max). The device has a data retention mode that guarantees that data will remain valid at minimum V_{CC} of 2.0V.

Functional Block Diagram



240572-1

Pin Connections



240572-2

Pin Names

A_0–A_{14}	Address
D_0–D_7	Data Input/Output
$\overline{CS}$	Chip Select
$\overline{WE}$	Write Enable
$\overline{OE}$	Output Enable
V_{CC}	Power
GND	Ground

Device Operation

The 51256S has two control inputs: Chip Select ($\overline{CS}$) and Write Enable ($\overline{WE}$). $\overline{CS}$ is the power control pin used for device operation. $\overline{WE}$ is the data control pin used to gate data at the I/O pins. Out Enable (OE) is used for precise control of the outputs.

Table 1. Mode Selection Truth Table

$\overline{CS}$	$\overline{WE}$	$\overline{OE}$	Mode	I/O	Power
H	X	X	Standby	High Z	Standby
L	X	H	Read	High Z	Active
L	H	L	Read	D_{OUT}	Active
L	L	X	Write	D_{IN}	Active

ABSOLUTE MAXIMUM RATINGS

Voltage on Any Pin
Relative to Ground (V_{IN}, V_{OUT}) ..... −0.3V to 7V
Storage Temperature (T_{STG}) .... −55°C to + 150°C
Power Dissipation (P_D) 1.0W
DC Continuous Output Current (I_{OS}) ..... 50 mA

**Notice: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.*

RECOMMENDED OPERATING CONDITIONS

Voltage referenced to V_{SS}, T_A = 0°C to 70°C

Parameter	Symbol	Min	Typ	Max	Unit
Supply Voltage	V_{CC}	4.5	5.0	5.5	V
Ground	V_{SS}	0	0	0	V
Input High Voltage	V_{IH}	2.2		V_{CC} + 0.5	V
Input Low Voltage	V_{IL}	−0.3		0.8	V

NOTE:
V_{IL} (Min) = −3.0V for 20 ns pulse.

CAPACITANCE T_A = 25°C, f = 1.0 MHz

Symbol	Parameter	Min	Max	Unit
C_{IN1}	Input Capacitance (V_{IN} = 0V)		8	pF
C_{OUT}	Output Capacitance (V_{OUT} = 0V)		10	pF

NOTE:
This parameter is sampled and not 100% tested.

2716
16K (2K x 8) UV ERASABLE PROM

- **Fast Access Time**
  - **—2716-1: 350 ns Max**
  - **—2716-2: 390 ns Max**
  - **—2716: 450 ns Max**
- **Single +5V Power Supply**
- **Low Power Dissipation**
  - **—Active Power: 525 mW Max**
  - **—Standby Power: 132 mW Max**
- **Pin Compatible to Intel "Universal Site" EPROMs**
- **Simple Programming Requirements**
  - **—Single Location Programming**
  - **—Programs with One 50 ms Pulse**
- **Inputs and Outputs TTL Compatible During Read and Program**
- **Completely Static**

The Intel 2716 is a 16,384-bit ultraviolet erasable and electrically programmable read-only memory (EPROM). The 2716 operates from a single 5-volt power supply, has a static standby mode, and features fast single-address programming. It makes designing with EPROMs fast, easy and economical.

The 2716, with its single 5-volt supply and with an access time up to 350 ns, is ideal for use with high-performance +5V microprocessors such as Intel's 8085 and 8086. Selected 2716-5s and 2716-6s are also available for slower speed applications. The 2716 also has a static standby mode which reduces power consumption without increasing access time. The maximum active power dissipation is 525 mW while the maximum standby power dissipation is only 132 mW, a 75% savings.

The 2716 uses a simple and fast method for programming—a single TTL-level pulse. There is no need for high voltage pulsing because all programming controls are handled by TTL signals. Programming of any location at any time—either individually, sequentially or at random is possible with the 2716's single-address programming. Total programming time for all 16,384 bits is only 100 seconds.

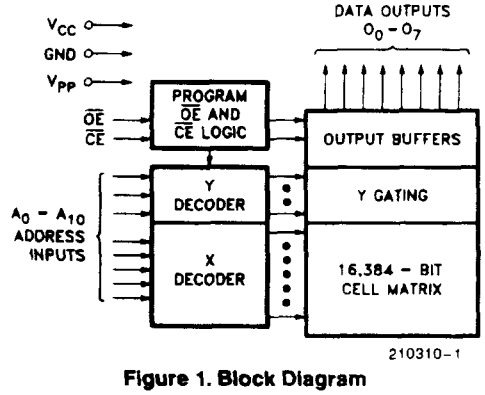
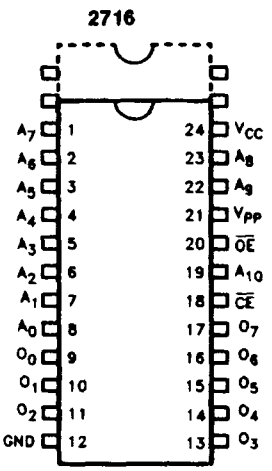


Figure 1. Block Diagram

Pin Names

A_0–A_{10}	Addresses
$\overline{CE}$	Chip Enable
$\overline{OE}$	Output Enable
O_0–O_7	Outputs

27512 27C512	27256 27C256	27128A 27C128	2764A 27C64 87C64	2732A
A_{15}	V_{PP}	V_{PP}	V_{PP}	
A_{12}	A_{12}	A_{12}	A_{12}	
A_7	A_7	A_7	A_7	A_7
A_6	A_6	A_6	A_6	A_6
A_5	A_5	A_5	A_5	A_5
A_4	A_4	A_4	A_4	A_4
A_3	A_3	A_3	A_3	A_3
A_2	A_2	A_2	A_2	A_2
A_1	A_1	A_1	A_1	A_1
A_0	A_0	A_0	A_0	A_0
O_0	O_0	O_0	O_0	O_0
O_1	O_1	O_1	O_1	O_1
O_2	O_2	O_2	O_2	O_2
GND	GND	GND	GND	GND



2732A	2764A 27C64 87C64	27128A 27C128	27256 27C256	27512 27C512
	V_{CC}	V_{CC}	V_{CC}	V_{CC}
	$\overline{PGM}$	$\overline{PGM}$	A_{14}	A_{14}
V_{CC}	N.C.	A_{13}	A_{13}	A_{13}
A_8	A_8	A_8	A_8	A_8
A_9	A_9	A_9	A_9	A_9
A_{11}	A_{11}	A_{11}	A_{11}	A_{11}
$\overline{OE}/V_{PP}$	$\overline{OE}$	$\overline{OE}$	$\overline{OE}$	$\overline{OE}/V_{PP}$
A_{10}	A_{10}	A_{10}	A_{10}	A_{10}
$\overline{CE}$	$\overline{OE}_{ALE}/\overline{CE}$	$\overline{CE}$	$\overline{CE}$	$\overline{CE}$
O_7	O_7	O_7	O_7	O_7
O_6	O_6	O_6	O_6	O_6
O_5	O_5	O_5	O_5	O_5
O_4	O_4	O_4	O_4	O_4
O_3	O_3	O_3	O_3	O_3

210310-2

NOTE:
Intel "Universal Site" compatible EPROM configurations are shown in the blocks adjacent to the 2716 pins.

Figure 2. Cerdip Pin Configuration

EXTENDED TEMPERATURE (EXPRESS) EPROMs

The Intel EXPRESS EPROM family is a series of electrically programmable read only memories which have received additional processing to enhance product characteristics. EXPRESS processing is available for several densities of EPROM, allowing the choice of appropriate memory size to match system applications. EXPRESS EPROM products are available with 168 ±8 hour, 125°C dynamic burn-in using Intel's standard bias configuration. This process exceeds or meets most industry specifications of burn-in. The standard EXPRESS EPROM operating temperature range is 0°C to 70°C. Extended operating temperature range (−40°C to +85°C) EXPRESS products are available. Like all Intel EPROMs, the EXPRESS EPROM family is inspected to 0.1% electrical AQL. This may allow the user to reduce or eliminate incoming inspection testing.

EXPRESS EPROM PRODUCT FAMILY

PRODUCT DEFINITONS

Type	Operating Temperature	Burn-In 125°C (hr)
Q	0°C to +70°C	168 ±8
I	−40°C to +85°C	44

EXPRESS OPTIONS

2716 Versions

Packaging Options	
Speed Versions	**Cerdip**
−1	Q
STD	Q, I

DEVICE OPERATION

The six modes of operation of the 2716 are listed in Table 1. It should be noted that inputs for all modes are TTL levels. The power supplies required are a +5V V_{CC} and a V_{PP}. The V_{PP} power supply must be at 25V during the three programming modes, and must be at 5V in the other three modes.

Read Mode

The 2716 has two control functions, both of which must be logically satisfied in order to obtain data at the outputs. Chip Enable ($\overline{CE}$) is the power control and should be used for device selection. Output Enable ($\overline{OE}$) is the output control and should be used to gate data from the output pins, independent of device selection. Assuming that addresses are stable, address access time (t_{ACC}) is equal to the delay from $\overline{CE}$ to output (t_{CE}). Data is available at the outputs t_{OE} after the falling edge of $\overline{OE}$, assuming that $\overline{CE}$ has been low and addresses have been stable for at least $t_{ACC}-t_{OE}$.

Standby Mode

The 2716 has a standby mode which reduces the maximum active power dissipation by 75%, from 525 mW to 132 mW. The 2716 is placed in the standby mode by applying a TTL-high signal to the $\overline{CE}$ input. When in standby mode, the outputs are in a high impedance state, independent of the $\overline{OE}$ input.

Output OR-Tieing

Because 2716s are usually used in larger memory arrays, Intel has provided a 2-line control function that accommodates this use of multiple memory connections. The two-line control function allows for:

a) the lowest possible memory power dissipation, and

b) complete assurance that output bus contention will not occur.

To use these two control lines most efficiently, $\overline{CE}$ (pin 18) should be decoded and used as the primary device selecting function, while $\overline{OE}$ (pin 20) should be made a common connection to all devices in the array and connected to the $\overline{READ}$ line from the system control bus. This assures that all deselected memory devices are in their low-power standby modes and that the output pins are active only when data is desired from a particular memory device.

Programming

Initially, and after each erasure, all bits of the 2716 are in the "1" state. Data is introduced by selectively programming "0's" into the desired bit locations. Although only "0's" will be programmed, both "1's" and "0's" can be presented in the data word. The only way to change a "0" to a "1" is by ultraviolet light erasure.

The 2716 is in the programming mode when the V_{PP} power supply is at 25V and $\overline{OE}$ is at V_{IH}. The data to be programmed is applied 8 bits in parallel to the data output pins. The levels required for the address and data inputs are TTL.

When the address and data are stable, a 50 ms, active-high, TTL program pulse is applied to the $\overline{CE}$ input. A pulse must be applied at each address location to be programmed. You can program any location at any time—either individually, sequentially, or at random. The program pulse has a maximum width of 55 ms. The 2716 must not be programmed with a DC signal applied to the $\overline{CE}$ input.

Table 1. Mode Selection

Pins / Mode	$\overline{CE}$ (18)	$\overline{OE}$ (20)	V_{PP} (21)	V_{CC} (24)	Outputs (9–11, 13–17)
Read	V_{IL}	V_{IL}	+5	+5	D_{OUT}
Output Disable	V_{IL}	V_{IH}	+5	+5	High Z
Standby	V_{IH}	X	+5	+5	High Z
Program	Pulsed V_{IL} to V_{IH}	V_{IH}	+25	+5	D_{IN}
Verify	V_{IL}	V_{IL}	+25	+5	D_{OUT}
Program Inhibit	V_{IL}	V_{IH}	+25	+5	High Z

NOTE:
1. X can be V_{IL} or V_{IH}.

Programming of multiple 2716s in parallel with the same data can be easily accomplished due to the simplicity of the programming requirements. Like inputs of the paralleled 2716s may be connected together when they are programmed with the same data. A low level TTL pulse applied to the $\overline{CE}$ input programs the paralleled 2716s.

Program Inhibit

Programming of multiple 2716s in parallel with different data is also easily accomplished. Except for $\overline{CE}$, all like inputs (including $\overline{OE}$) of the parallel 2716s may be common. A TTL-level program pulse applied to a 2716's $\overline{CE}$ input with V_{PP} at 25V will program that 2716. A low-level $\overline{CE}$ input inhibits the other 2716 from being programmed.

Verify

A verify should be performed on the programmed bits to determine that they were correctly programmed. The verify may be performed with V_{PP} at 25V. Except during programming and program verify, V_{PP} must be at 5V.

ERASURE CHARACTERISTICS

The erasure characteristics of the 2716 are such that erasure begins to occur upon exposure to light with wavelengths shorter than aproximately 4000 Angstroms (Å). It should be noted that sunlight and certain types of fluorescent lamps have wavelengths in the 3000–4000Å range. Data show that constant exposure to room-level fluorescent lighting could erase the typical 2716 in approximately 3 years, while it would take approximately 1 week to cause erasure when exposed to direct sunlight. If the 2716 is to be exposed to these types of lighting conditions for extended periods of time, opaque labels should be placed over the window to prevent unintentional erasure.

The recommended erasure procedure for the 2716 is exposure to shortwave ultraviolet light which has a wavelength of 2537 Angstroms (Å). The integrated dose (i.e., UV intensity × exposure time) for erasure should be a minimum of 15 Ws/cm^2. The erasure time with this dosage is approximately 15 to 20 minutes using an ultraviolet lamp with a 12000 $\mu W/cm^2$ power rating. The 2716 should be placed within 1 inch of the lamp tubes during erasure.

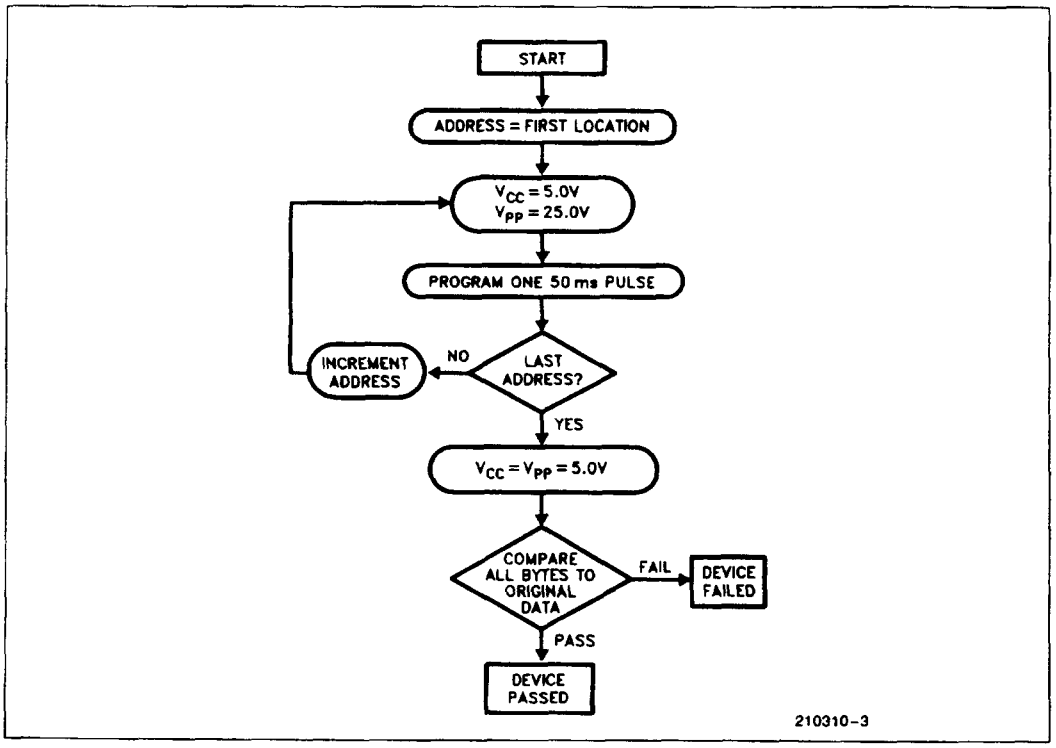


Figure 3. Standard Programming Flowchart

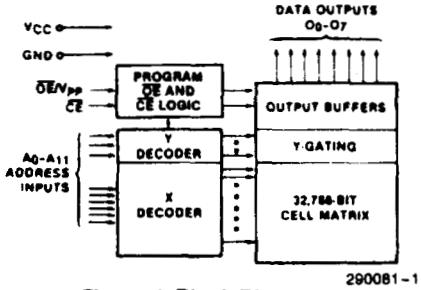
2732A
32K (4K x 8) UV ERASABLE PROMS

- **200 ns (2732A-2) Maximum Access Time ... HMOS*-E Technology**
- **Compatible with High-Speed Microcontrollers and Microprocessors ... Zero WAIT State**
- **Two Line Control**
- **10% V_{CC} Tolerance Available**
- **Low Current Requirement**
  −100 mA Active
  −35 mA Standby
- **Int$_e$ligent Identifier™ Mode**
  −Automatic Programming Operation
- **Industry Standard Pinout ... JEDEC Approved 24 Pin Ceramic Package**
  (See Packaging Spec. Order # 231369)

The Intel 2732A is a 5V-only, 32,768-bit ultraviolet erasable (cerdip) Electrically Programmable Read-Only Memory (EPROM). The standard 2732A access time is 250 ns with speed selection (2732A-2) available at 200 ns. The access time is compatible with high performance microprocessors such as the 8 MHz iAPX 186. In these systems, the 2732A allows the microprocessor to operate without the addition of WAIT states.

An important 2732A feature is Output Enable ($\overline{OE}$) which is separate from the Chip Enable ($\overline{CE}$) control. The $\overline{OE}$ control eliminates bus contention in microprocessor systems. The $\overline{CE}$ is used by the 2732A to place it in a standby mode ($\overline{CE}$ = V_{IH}) which reduces power consumption without increasing access time. The standby mode reduces the current requirement by 65%; the maximum active current is reduced from 100 mA to a standby current of 35 mA.

*HMOS is a patented process of Intel Corporation.



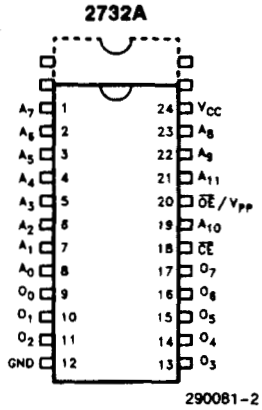
290081-1

Figure 1. Block Diagram

Pin Names

A_0-A_{11}	Addresses
$\overline{CE}$	Chip Enable
$\overline{OE}/V_{PP}$	Output Enable/V_{PP}
O_0-O_7	Outputs

27512 27C512	27256 27C256	27128A 27C128	2764A 27C64 87C64	2716
A_{15}	V_{PP}	V_{PP}	V_{PP}	
A_{12}	A_{12}	A_{12}	A_{12}	
A_7	A_7	A_7	A_7	A_7
A_6	A_6	A_6	A_6	A_6
A_5	A_5	A_5	A_5	A_5
A_4	A_4	A_4	A_4	A_4
A_3	A_3	A_3	A_3	A_3
A_2	A_2	A_2	A_2	A_2
A_1	A_1	A_1	A_1	A_1
A_0	A_0	A_0	A_0	A_0
O_0	O_0	O_0	O_0	O_0
O_1	O_1	O_1	O_1	O_1
O_2	O_2	O_2	O_2	O_2
GND	GND	GND	GND	GND



290081-2

2716	2764 2764A 87C64	27128A 27C128	27256 27C256	27512 27C512
	V_{CC} / $\overline{PGM}$	V_{CC} / $\overline{PGM}$	V_{CC} / A_{14}	V_{CC} / A_{14}
V_{CC}	N.C.	A_{13}	A_{13}	A_{13}
A_8	A_8	A_8	A_8	A_8
A_9	A_9	A_9	A_9	A_9
V_{PP}	A_{11}	A_{11}	A_{11}	A_{11}
$\overline{OE}$	$\overline{OE}$	$\overline{OE}$	$\overline{OE}$	$\overline{OE}/V_{PP}$
A_{10}	A_{10}	A_{10}	A_{10}	A_{10}
$\overline{CE}$	$\overline{CE}$ / ALE/$\overline{CE}$	$\overline{CE}$	$\overline{CE}$	$\overline{CE}$
O_7	O_7	O_7	O_7	O_7
O_6	O_6	O_6	O_6	O_6
O_5	O_5	O_5	O_5	O_5
O_4	O_4	O_4	O_4	O_4
O_3	O_3	O_3	O_3	O_3

NOTE:
Intel "Universal Site" compatible EPROM configurations are shown in the blocks adjacent to the 2732A pins.

Figure 2. Cerdip Pin Configuration

EXTENDED TEMPERATURE (EXPRESS) EPROMs

The Intel EXPRESS EPROM family is a series of electrically programmable read only memories which have received additional processing to enhance product characteristics. EXPRESS processing is available for several densities of EPROM, allowing the choice of appropriate memory size to match system applications. EXPRESS EPROM products are available with 168 ±8 hour, 125°C dynamic burn-in using Intel's standard bias configuration. This process exceeds or meets most industry specifications of burn-in. The standard EXPRESS EPROM operating temperature range is 0°C to 70°C. Extended operating temperature range (−40°C to +85°C) EXPRESS products are available. Like all Intel EPROMs, the EXPRESS EPROM family is inspected to 0.1% electrical AQL. This may allow the user to reduce or eliminate incoming inspection testing.

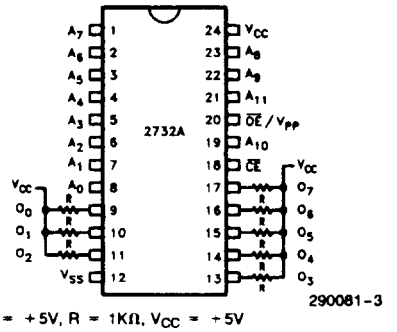
READ OPERATION

D.C. CHARACTERISTICS

Electrical Parameters of EXPRESS EPROM products are identical to standard EPROM parameters except for:

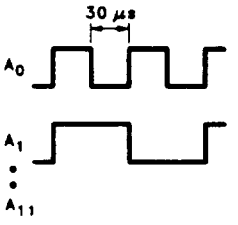
Symbol	Parameter	TD2732A LD2732A		Test Conditions
		Min	Max	
I_{SB}	V_{CC} Standby Current (mA)		45	$\overline{CE} = V_{IH}$, $\overline{OE} = V_{IL}$
I_{CC_1}(1)	V_{CC} Active Current (mA)		150	$\overline{OE} = \overline{CE} = V_{IL}$
	V_{CC} Active Current at High Temperature (mA)		125	$\overline{OE} = \overline{CE} = V_{IL}$, $V_{PP} = V_{CC}$, $T_{Ambient} = 85°C$

NOTE:
1. Maximum current value is with outputs O_0 to O_7 unloaded.



290081-3

$\overline{OE}/V_{PP}$ = +5V, R = 1KΩ, V_{CC} = +5V
V_{SS} = GND, $\overline{CE}$ = GND



290081-4

Binary Sequence from A_0 to A_{11}

EXPRESS EPROM PRODUCT FAMILY

PRODUCT DEFINITONS

Type	Operating Temperature	Burn-In 125°C (hr)
Q	0°C to +70°C	168 ±8
T	−40°C to +85°C	None
L	−40°C to +85°C	168 ±8

EXPRESS OPTIONS

2732A Versions

Packaging Options	
Speed Versions	Cerdip
−2	Q
−25	Q, T, L

every eight devices. The bulk capacitor should be located near where the power supply is connected to the array. The purpose of the bulk capacitor is to overcome the voltage droop caused by the inductive effects of PC board traces.

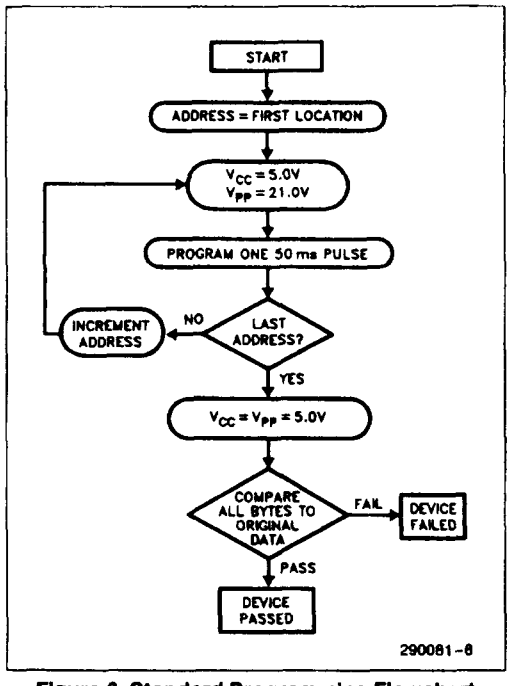


Figure 3. Standard Programming Flowchart

PROGRAMMING MODES

CAUTION: Exceeding 22V on $\overline{OE}/V_{PP}$ will permanently damage the device.

Initially, and after each erasure (cerdip EPROMs), all bits of the EPROM are in the "1" state. Data is introduced by selectively programming "0s" into the bit locations. Although only "0s" will be programmed, both "1s" and "0s" can be present in the data word. The only way to change a "0" to a "1" in cerdip EPROMs is by ultraviolet light erasure.

The device is in the programming mode when the $\overline{OE}/V_{PP}$ input is at 21V. It is required that a 0.1 μF capacitor be placed across $\overline{OE}/V_{PP}$ and ground to suppress spurious voltage transients which may damage the device. The data to be programmed is applied 8 bits in parallel to the data output pins. The levels required for the address and data inputs are TTL.

When the address and data are stable, a 20 ms (50 ms typical) active low, TTL program pulse is applied to the $\overline{CE}$ input. A program pulse must be applied at each address location to be programmed (see Figure 3). Any location can be programmed at any time—either individually, sequentially, or at random. The program pulse has a maximum width of 55 ms. The EPROM must not be programmed with a DC signal applied to the $\overline{CE}$ input.

Programming of multiple 2732As in parallel with the same data can be easily accomplished due to the simplicity of the programming requirements. Like inputs of the paralleled 2732As may be connected together when they are programmed with the same data. A low level TTL pulse applied to the $\overline{CE}$ input programs the paralleled 2732As.

Program Inhibit

Programming of multiple EPROMs in parallel with different data is easily accomplished by using the Program Inhibit mode. A high level $\overline{CE}$ input inhibits the other EPROMs from being programmed. Except for $\overline{CE}$, all like inputs (including $\overline{OE}/V_{PP}$) of the parallel EPROMs may be common. A TTL low level pulse applied to the $\overline{CE}$ input with $\overline{OE}/V_{PP}$ at 21V will program that selected device.

Program Verify

A verify (Read) should be performed on the programmed bits to determine that they have been correctly programmed. The verify is performed with $\overline{OE}/V_{PP}$ and $\overline{CE}$ at V_{IL}. Data should be verified t_{DV} after the falling edge of $\overline{CE}$.

int$_e$ligent Identifier™ Mode

The int$_e$ligent Identifier Mode allows the reading out of a binary code from an EPROM that will identify its manufacturer and type. This mode is intended for use by programming equipment for the purpose of automatically matching the device to be programmed with its corresponding programming algorithm. This mode is functional in the 25°C ±5°C ambient temperature range that is required when programming the device.

To activate this mode, the programming equipment must force 11.5V to 12.5V on address line A9 of the EPROM. Two identifier bytes may then be sequenced from the device outputs by toggling address line A0 from V_{IL} to V_{IH}. All other address lines must be held at V_{IL} during the int$_e$ligent Identifier Mode.

Byte 0 (A0 = V_{IL}) represents the manufacturer code and byte 1 (A0 = V_{IH}) the device identifier code. These two identifier bytes are given in Table 1.

2764A
64K (8K x 8)
UV ERASABLE PROMs

- **Fast Access Time—HMOS* II E — 180 ns Cerdip D2764A-1**
- **Moisture Resistant**
- **Two-line Control**
- **Int$_e$ligent Identifier™ Mode**
- **Industry Standard Pinout . . . JEDEC Approved . . . 28 Lead Package**
  (See Packaging Spec, Order # 231369)

The Intel 2764A is a 5V only, 65,536-bit electrically programmable read-only memory (EPROM). The 2764A is fabricated with Intel's HMOSII-E technology which significantly reduces die size and greatly improves the device's performance, power consumption, reliability and producibility.

The 2764A provides access times to 180 ns (2764A-1). This is compatible with high-performance microprocessors, such as Intel's 8 MHz iAPX 186 allowing full speed operation without the addition of WAIT states. The 2764A is also directly compatible with the 12 MHz 8051 family.

Two-line control and JEDEC-approved, 28 pin packaging are standard features of Intel higher density EPROMs. This assures easy microprocessor interfacing and minimum design efforts when upgrading, adding, or choosing between non-volatile memory alternatives.

*HMOS is a patented process of Intel Corporation.

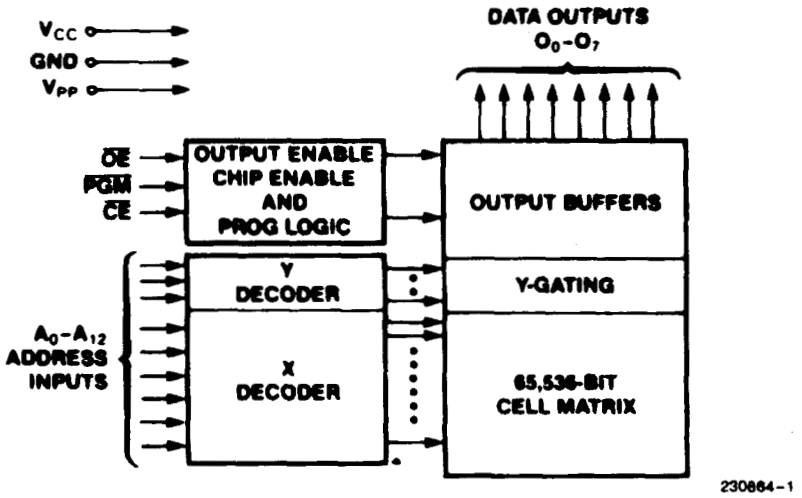
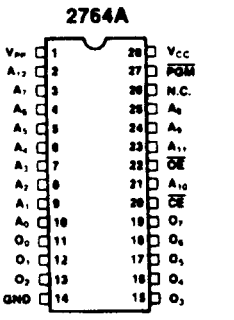


Figure 1. Block Diagram

Pin Names

A_0–A_{12}	Addresses
$\overline{CE}$	Chip Enable
$\overline{OE}$	Output Enable
O_0–O_7	Outputs
$\overline{PGM}$	Program
N.C.	No Connect

27512 27C512	27256 27C256	27128A 27C128	2732A	2716
A_{15}	V_{PP}	V_{PP}		
A_{12}	A_{12}	A_{12}		
A_7	A_7	A_7	A_7	A_7
A_6	A_6	A_6	A_6	A_6
A_5	A_5	A_5	A_5	A_5
A_4	A_4	A_4	A_4	A_4
A_3	A_3	A_3	A_3	A_3
A_2	A_2	A_2	A_2	A_2
A_1	A_1	A_1	A_1	A_1
A_0	A_0	A_0	A_0	A_0
O_0	O_0	O_0	O_0	O_0
O_1	O_1	O_1	O_1	O_1
O_2	O_2	O_2	O_2	O_2
GND	GND	GND	GND	GND



2716	2732A	27128A 27C128	27256 27C256	27512 27C512
		V_{CC}	V_{CC}	V_{CC}
		$\overline{PGM}$	A_{14}	A_{14}
V_{CC}	V_{CC}	A_{13}	A_{13}	A_{13}
A_8	A_8	A_8	A_8	A_8
A_9	A_9	A_9	A_9	A_9
V_{PP}	A_{11}	A_{11}	A_{11}	A_{11}
$\overline{OE}$	$\overline{OE}/V_{PP}$	$\overline{OE}$	$\overline{OE}$	$\overline{OE}/V_{PP}$
A_{10}	A_{10}	A_{10}	A_{10}	A_{10}
$\overline{CE}$	$\overline{CE}$	$\overline{CE}$	$\overline{CE}$	$\overline{CE}$
O_7	O_7	O_7	O_7	O_7
O_6	O_6	O_6	O_6	O_6
O_5	O_5	O_5	O_5	O_5
O_4	O_4	O_4	O_4	O_4
O_3	O_3	O_3	O_3	O_3

230864-2

NOTE:
Intel "Universal Site"-Compatible EPROM pin configurations are shown in the blocks adjacent to the 2764A pins.

Figure 2. Cerdip Pin Configuration

EXTENDED TEMPERATURE (EXPRESS) EPROMs

The Intel EXPRESS EPROM family is a series of electrically programmable read only memories which have received additional processing to enhance product characteristics. EXPRESS processing is available for several densities of EPROM, allowing the choice of appropriate memory size to match system applications. EXPRESS EPROM products are available with 168 ±8 hour, 125°C dynamic burn-in using Intel's standard bias configuration. This process exceeds or meets most industry specifications of burn-in. The standard EXPRESS EPROM operating temperature range is 0°C to 70°C. Extended operating temperature range (−40°C to +85°C) EXPRESS products are available. Like all Intel EPROMs, the EXPRESS EPROM family is inspected to 0.1% electrical AQL. This may allow the user to reduce or eliminate incoming inspection testing.

EXPRESS EPROM PRODUCT FAMILY

PRODUCT DEFINITIONS

Type	Operating Temperature	Burn-in 125°C (hr)
Q	0°C to +70°C	168 ±8
T	−40°C to +85°C	None
L	−40°C to +85°C	168 ±8

EXPRESS OPTIONS

2764A VERSIONS

Packaging Options	
Speed Versions	Cerdip
−20	Q, T, L

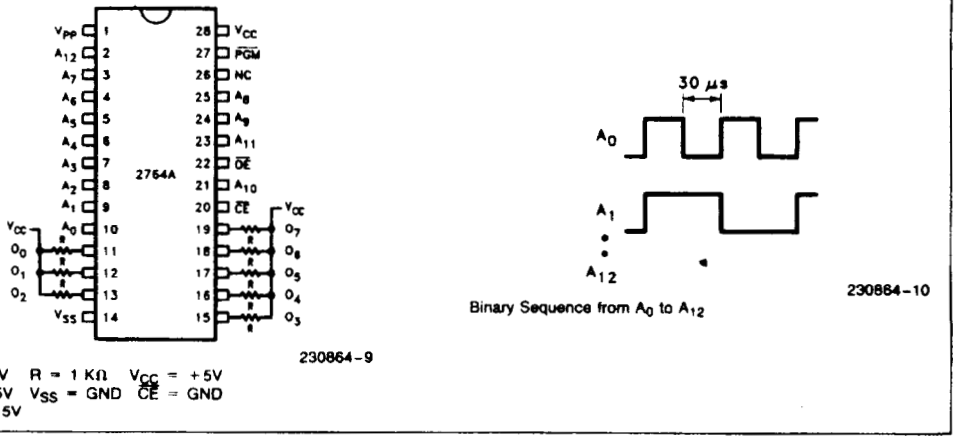
READ OPERATION

D.C. CHARACTERISTICS

Electrical parameters of EXPRESS EPROM products are identical to standard EPROM parameters except for:

Symbol	Parameter	TD2764A LD2764A		Test Conditions
		Min	Max	
I_{SB}	V_{CC} Standby Current (mA)		40	$\overline{CE} = V_{IH}$, $\overline{OE} = V_{IL}$
I_{CC_1}(1)	V_{CC} Active Current (mA)		100	$\overline{OE} = \overline{CE} = V_{IL}$
	V_{CC} Active Current at High Temperature (mA)		75	$\overline{OE} = \overline{CE} = V_{IL}$ $V_{PP} = V_{CC}$, $T_{Ambient} = 85°C$

NOTE:
1. The maximum current value is with outputs O_0 to O_7 unloaded.



Burn-in Bias and Timing Diagrams

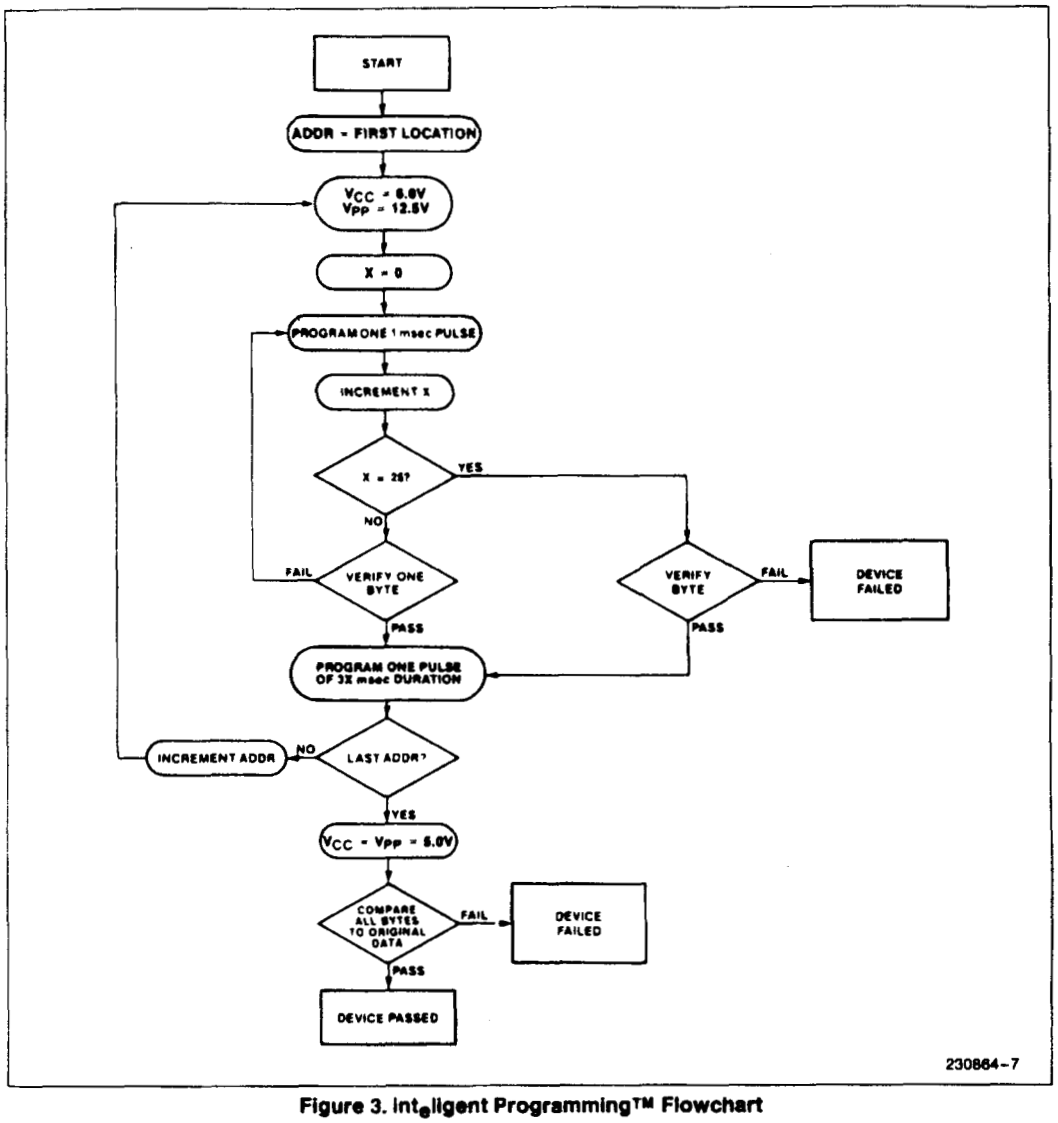


Figure 3. int$_e$ligent Programming™ Flowchart

int$_e$ligent Programming™ Algorithm

The int$_e$ligent Programming Algorithm, a standard in the industry for the past few years, is required for all of Intel's 12.5V CERDIP EPROMs. Plastic EPROMs may also be programmed using this method. A flowchart of the int$_e$ligent Programming Algorithm is shown in Figure 3.

The int$_e$ligent Programming Algorithm utilizes two different pulse types: initial and overprogram. The duration of the initial $\overline{PGM}$ pulse(s) is one millisecond, which will then be followed by a longer overprogram pulse of length 3X msec. X is an iteration counter and is equal to the number of the initial one millisecond pulses applied to a particular location, before a correct verify occurs. Up to 25 one-millisecond pulses per byte are provided for before the overprogram pulse is applied.

The entire sequence of program pulses and byte verifications is performed at V_{CC} = 6.0V and V_{PP} = 12.5V. When the int$_e$ligent Programming cycle has been completed, all bytes should be compared to the original data with V_{CC} = V_{PP} = 5.0V.

27C64/87C64
64K (8K x 8) CHMOS PRODUCTION AND UV ERASABLE PROMS

- **CHMOS Microcontroller and Microprocessor Compatible**
  - **87C64-Integrated Address Latch**
  - **Universal 28 Pin Memory Site, 2-line Control**
- **Low Power Consumption**
  - **100 μA Maximum Standby Current**
- **Noise Immunity Features**
  - **± 10% V_{CC} Tolerance**
  - **Maximum Latch-up Immunity Through EPI Processing**
- **High Performance Speeds**
  - **150 ns Maximum Access Time**
- **New Quick-Pulse Programming™ Algorithm (1 second programming)**
- **Available in 28-Pin Cerdip and Plastic DIP Package and 32-Lead PLCC Package.**

(See Packaging Spec, Order #231369)

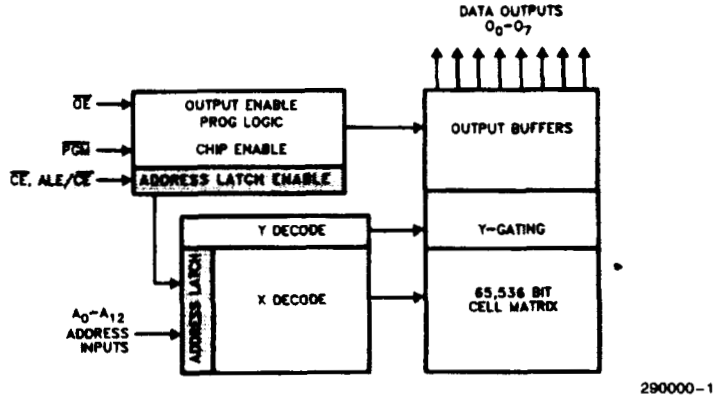
Intel's 27C64 and 87C64 CHMOS EPROMs are 64K bit 5V only memories organized as 8192 words of 8 bits. They employ advanced CHMOS*II-E circuitry for systems requiring low power, high performance speeds, and immunity to noise. The 87C64 has been optimized for multiplexed bus microcontroller and microprocessor compatibility while the 27C64 has a non-multiplexed addressing interface and is plug compatible with the standard Intel 2764A (HMOS II-E).

The 27C64 and 87C64 are offered in both a ceramic DIP, Plastic DIP, and Plastic Leaded Chip Carrier (PLCC) Packages. Cerdip packages provide flexibility in prototyping and R&D environments, whereas Plastic DIP and PLCC EPROMs provide optimum cost effectiveness in production environments. A new Quick-Pulse Programming™ Algorithm is employed which can speed up programming by as much as one hundred times.

The 87C64 incorporates an address latch on the address pins to minimize chip count in multiplexed bus systems. Designers can eliminate an external address latch by tieing address and data pins of the 87C64 directy to the processor's multiplexed address/data pins. On the falling edge of the ALE input (ALE/$\overline{CE}$), address information at the address inputs (A_0–A_{12}) of the 87C64 is latched internally. The address inputs are then ignored as data information is passed on the same bus.

The highest degree of protection against latch-up is achieved through Intel's unique EPI processing. Prevention of latch-up is provided for stresses up to 100 mA on address and data pins from −1V to V_{CC} + 1V.

*HMOS and CHMOS are patented processes of Intel Corporation.



Shaded Areas represent the 87C64 version

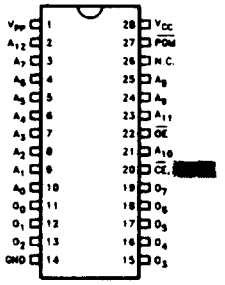
Figure 1. Block Diagram

Pin Names

A_0-A_{12}	ADDRESSES
O_0-O_7	OUTPUTS
$\overline{OE}$	OUTPUT ENABLE
$\overline{CE}$	CHIP ENABLE
ALE/$\overline{CE}$	ADDRESS LATCH ENABLE /CHIP ENABLE
$\overline{PGM}$	PROGRAM STROBE
N.C.	NO CONNECT
D.U.	DON'T USE

27C64/87C64
P27C64/P87C64

27512 27C512	27256 27C256	27128A 27C128	2732A	2716
A_{15}	V_{PP}	V_{PP}		
A_{12}	A_{12}	A_{12}		
A_7	A_7	A_7	A_7	A_7
A_6	A_6	A_6	A_6	A_6
A_5	A_5	A_5	A_5	A_5
A_4	A_4	A_4	A_4	A_4
A_3	A_3	A_3	A_3	A_3
A_2	A_2	A_2	A_2	A_2
A_1	A_1	A_1	A_1	A_1
A_0	A_0	A_0	A_0	A_0
O_0	O_0	O_0	O_0	O_0
O_1	O_1	O_1	O_1	O_1
O_2	O_2	O_2	O_2	O_2
GND	GND	GND	GND	GND

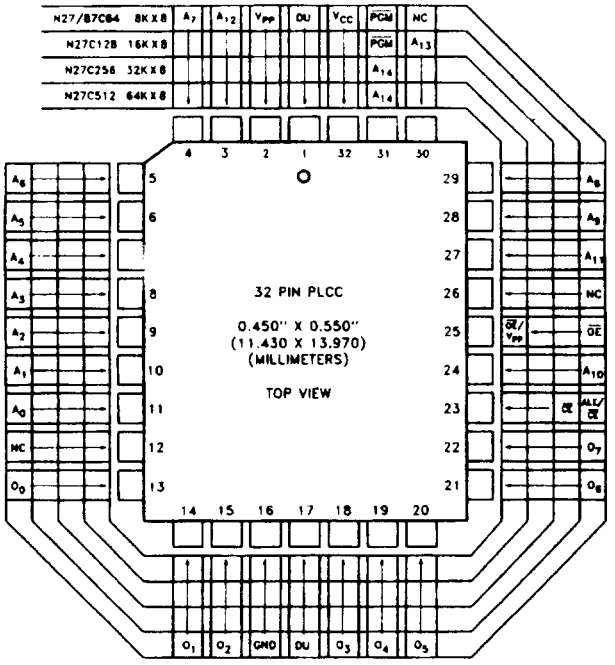


2716	2732A	27128A 27C128	27256 27C256	27512 27C512
		V_{CC} / $\overline{PGM}$	V_{CC}	V_{CC}
		A_{13}	A_{14}	A_{14}
V_{CC}	V_{CC}	A_8	A_{13}	A_{13}
A_8	A_8	A_9	A_8	A_8
A_9	A_9	A_{11}	A_9	A_9
V_{PP}	A_{11}	$\overline{OE}$	A_{11}	A_{11}
$\overline{OE}$	$\overline{OE}/V_{PP}$	A_{10}	$\overline{OE}$	$\overline{OE}/V_{PP}$
A_{10}	A_{10}	$\overline{CE}$	A_{10}	A_{10}
$\overline{CE}$	$\overline{CE}$	O_7	$\overline{CE}$	$\overline{CE}$
O_7	O_7	O_6	O_7	O_7
O_6	O_6	O_5	O_6	O_6
O_5	O_5	O_4	O_5	O_5
O_4	O_4	O_3	O_4	O_4
O_3	O_3		O_3	O_3

290000-2

NOTE:
Intel "Universal Site" Compatible EPROM Pin Configurations are shown in the adjacent blocks to 27C64 Pins.
Shaded Areas represent the 87C64 version

Figure 2. Pin Configuration



290000-11

Figure 3. PLCC(N) Lead Configuration

Extended Temperature (Express) EPROMs

The Intel EXPRESS EPROM family is a series of electrically programmable read only memories which have received additional processing to enhance product characteristics. EXPRESS processing is available for several densities of EPROM, allowing the choice of appropriate memory size to match system applications.

EXPRESS EPROM products are available with 168 ±8 hour, 125°C dynamic burn-in using Intel's standard bias configuration. This process exceeds or meets most industry specifications of burn-in. The standard EXPRESS EPROM operating temperature range is 0°C to 70°C. Extended operating temperature range (−40°C to +85°C) EXPRESS products are also available. Like all Intel EPROMs, the EXPRESS EPROM family is inspected to 0.1% electrical AQL. This may allow the user to reduce or eliminate incoming inspection testing.

EXPRESS EPROM Product Family

PRODUCT DEFINITIONS

Type	Operating Temperature (°C)	Burn-In 125°C (hr)
Q	0 to +70	168 ±8
T	−40 to +85	NONE
L	−40 to +85	168 ±8

EXPRESS Options

27C64/87C64 Versions

Packaging Options		
Speed Versions	**Cerdip**	**PLCC**
−1	Q (27)	
−20	T, L, Q	T

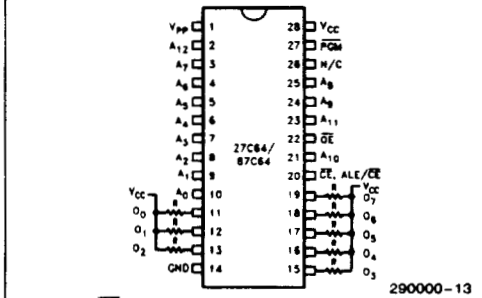
READ OPERATION

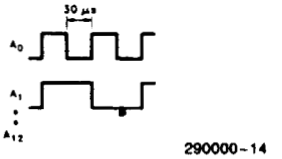
D.C. CHARACTERISTICS

Electrical Parame s of EXPRESS EPROM products are identical to standard EPROM parameters except for:

Symbol	Parameter		27C64 87C64 Min	27C64 87C64 Max	Test Conditions
I_{SB}	V_{CC} Standby Current (mA)	CMOS		0.1	$\overline{CE} = V_{CC}$, $\overline{OE} = V_{IL}$
		TTL		1.0	$\overline{CE} = V_{IH}$, $\overline{OE} = V_{IL}$
I_{CC_1}(1)	V_{CC} Active Current (mA)	TTL		20, 30	$\overline{OE} = \overline{CE} = V_{IL}$
	V_{CC} Active Current at High Temperature	TTL		20, 30	$\overline{OE} = \overline{CE} = V_{IL}$ $V_{PP} = V_{CC}$, $T_{ambient} = 85°C$

NOTE:
1. See notes 4 and 6 of Read Operation D.C. Characteristics.





Burn-In Bias and Timing Diagrams

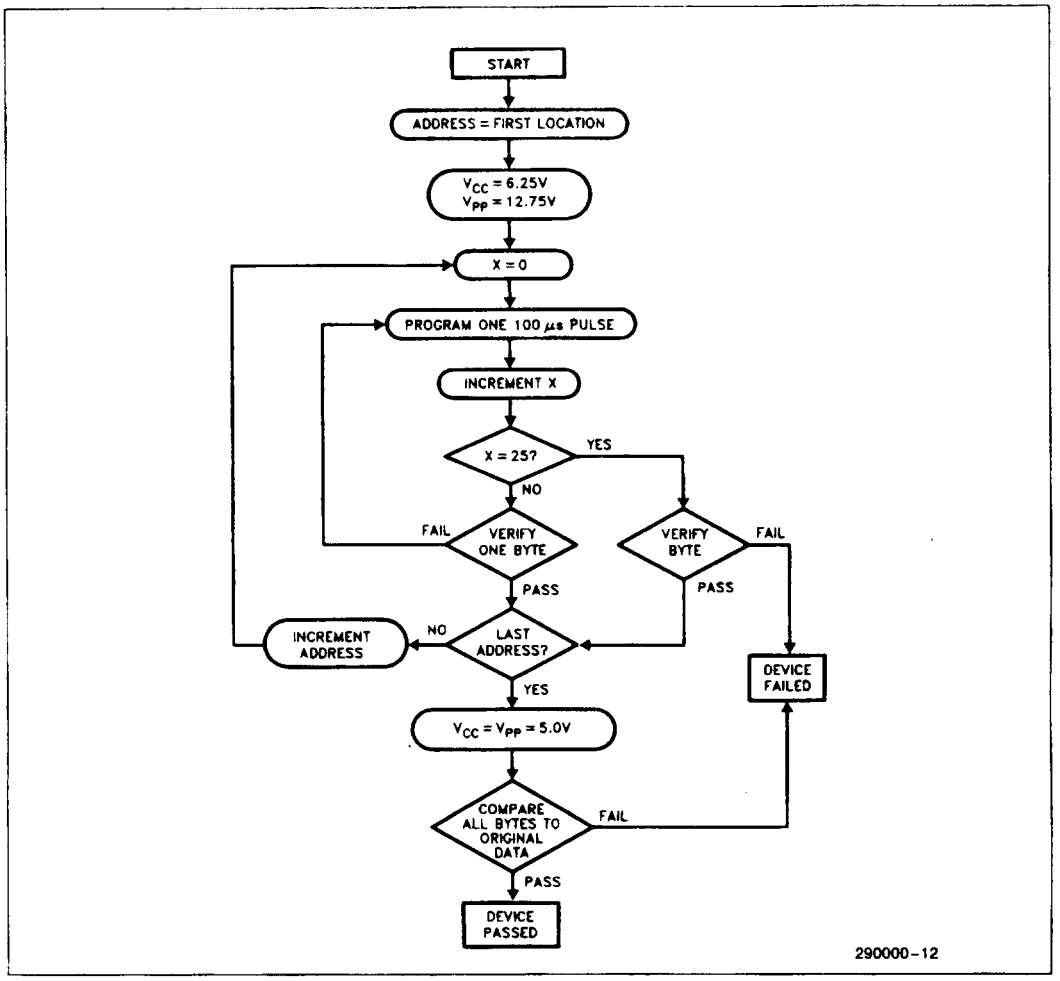


Figure 5. Quick-Pulse Programming™ Algorithm

Quick-Pulse Programming™ Algorithm

Intel's 27C64 and 87C64 EPROMs can now be programmed using the Quick-Pulse Programming Algorithm, developed by Intel to substantially reduce the throughput time in the production environment. This algorithm allows these devices to be programmed in under one second, almost a hundred fold improvement over previous algorithms. Actual programming time is a function of the PROM programmer being used.

The Quick-Pulse Programming Algorithm uses initial pulses of 100 microseconds followed by a byte verification to determine when the address byte has been successfully programmed. Up to 25 100 µs pulses per byte are provided before a failure is recognized. A flowchart of the Quick-Pulse Programming Algorithm is shown in Figure 5.

For the Quick Pulse Programming Algorithm, the entire sequence of programming pulses and byte verifications is performed at V_{CC} = 6.25V and V_{PP} at 12.75V. When programming of the EPROM has been completed, all bytes should be compared to the original data with V_{CC} = V_{PP} = 5.0V.

27128A
128K (16K x 8) PRODUCTION AND UV ERASABLE PROMs

- **Fast 150 nsec Access Time**
  — HMOS* II-E Technology
- **Low Power**
  — 100 mA Maximum Active
  — 40 mA Maximum Standby
- **int$_e$ligent Identifier™ Mode**
  — Automated Programming Operations
- **New Quick-Pulse Programming™ Algorithm**
  — Used on Plastic DIP
  — int$_e$ligent Programming™ Algorithm Compatible
- **± 10% V_{CC} Tolerance Available**
- **Available in 28-Pin Cerdip and Plastic Packages**
  (See Packaging Spec, Order # 231369)

The Intel 27128A is a 5V only, 131,072-bit ultraviolet erasable and electrically programmable read-only memory (EPROM). The 27128A is fabricated with Intel's HMOSII-E technology which significantly reduces die size and greatly improves the device's performance, reliability and manufacturability.

The 27128A is currently available in two different package types. CERDIP packages provide flexibility in prototyping and R&D environments where reprogrammability is required. Plastic DIP EPROMs provide optimum cost effectiveness in production environments.

Intel's new Quick-Pulse Programming Algorithm enables these Plastic EPROMs to be programmed within two seconds. Programming equipment that takes advantage of this innovation will electronically identify the EPROM with the help of the int$_e$ligent Identifier and rapidly program it using a superior programming method. The int$_e$ligent Programming Algorithm may be utilized in the absence of such equipment and is used to program CERDIP devices.

The 27128A is available in fast access times including 150 ns (27128A-1). This ensures compatibility with high-performance microprocessors, such as Intel's 8 MHz 80186 allowing full speed operation without the addition of WAIT states. The 27128A is also directly compatible with the 12 MHz 8051 family.

*HMOS is a patented process of Intel Corporation.

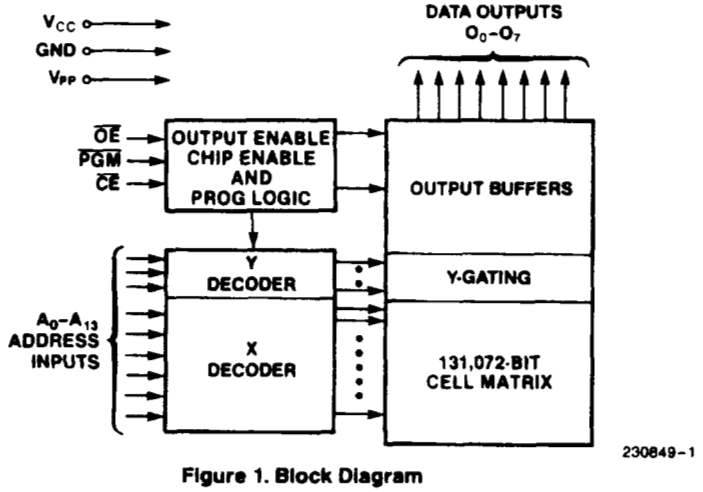


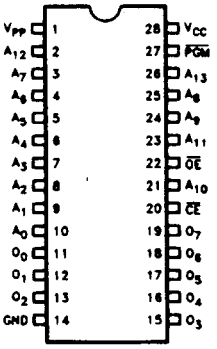
Figure 1. Block Diagram

Pin Names

A_0–A_{13}	ADDRESSES
$\overline{CE}$	CHIP ENABLE
$\overline{OE}$	OUTPUT ENABLE
O_0-O_7	OUTPUTS
$\overline{PGM}$	PROGRAM
N.C.	NO INTERNAL CONNECT

27512 27C512	27256 27C256	2764A 27C64 87C64	2732A	2716
A_{15}	V_{PP}	V_{PP}		
A_{12}	A_{12}	A_{12}		
A_7	A_7	A_7	A_7	A_7
A_6	A_6	A_6	A_6	A_6
A_5	A_5	A_5	A_5	A_5
A_4	A_4	A_4	A_4	A_4
A_3	A_3	A_3	A_3	A_3
A_2	A_2	A_2	A_2	A_2
A_1	A_1	A_1	A_1	A_1
A_0	A_0	A_0	A_0	A_0
O_0	O_0	O_0	O_0	O_0
O_1	O_1	O_1	O_1	O_1
O_2	O_2	O_2	O_2	O_2
GND	GND	GND	GND	GND

27128A
P27128A



230849-2

2716	2732A	2764A 27C64 87C64	27256 27C256	27512 27C512
		V_{CC}	V_{CC}	V_{CC}
		$\overline{PGM}$	A_{14}	A_{14}
V_{CC}	V_{CC}	N.C.	A_{13}	A_{13}
A_8	A_8	A_8	A_8	A_8
A_9	A_9	A_9	A_9	A_9
V_{PP}	A_{11}	A_{11}	A_{11}	A_{11}
$\overline{OE}$	$\overline{OE}/V_{PP}$	$\overline{OE}$	$\overline{OE}$	$\overline{OE}/V_{PP}$
A_{10}	A_{10}	A_{10}	A_{10}	A_{10}
$\overline{CE}$	$\overline{CE}$	$\overline{CE}$ ALE/$\overline{CE}$	$\overline{CE}$ ALE/$\overline{CE}$	$\overline{CE}$
O_7	O_7	O_7	O_7	O_7
O_6	O_6	O_6	O_6	O_6
O_5	O_5	O_5	O_5	O_5
O_4	O_4	O_4	O_4	O_4
O_3	O_3	O_3	O_3	O_3

NOTE: Intel "Universal Site"—Compatible EPROM Pin Configurations are Shown in the Blocks Adjacent to the 27128A Pins

Figure 2. Cerdip(D)/Plastic(P) DIP Pin Configurations

EXTENDED TEMPERATURE (EXPRESS) EPROMS

The Intel EXPRESS EPROM family is a series of electrically programmable read only memories which have received additional processing to enhance product characteristics. EXPRESS processing is available for several densities of EPROM, allowing the choice of appropriate memory size to match system applications. EXPRESS EPROM products are available with 168 ±8 hour, 125°C dynamic burn-in using Intel's standard bias configuration. This process exceeds or meets most industry specifications of burn-in. The standard EXPRESS EPROM operating temperature range is 0°C to 70°C. Extended operating temperature range (−40°C to +85°C) EXPRESS products are available. Like all Intel EPROMs, the EXPRESS EPROM family is inspected to 0.1% electrical AQL. This may allow the user to reduce or eliminate incoming inspection testing.

EXPRESS EPROM PRODUCT FAMILY

PRODUCT DEFINITIONS

Type	Operating Temperature	Burn-In 125°C (hr)
Q	0°C to +70°C	168 ±8
T	−40°C to +85°C	None
L	−40°C to +85°C	168 ±8

EXPRESS OPTIONS

27128A Versions

Packaging Options	
Speed Versions	Cerdip
-20	T, L, Q

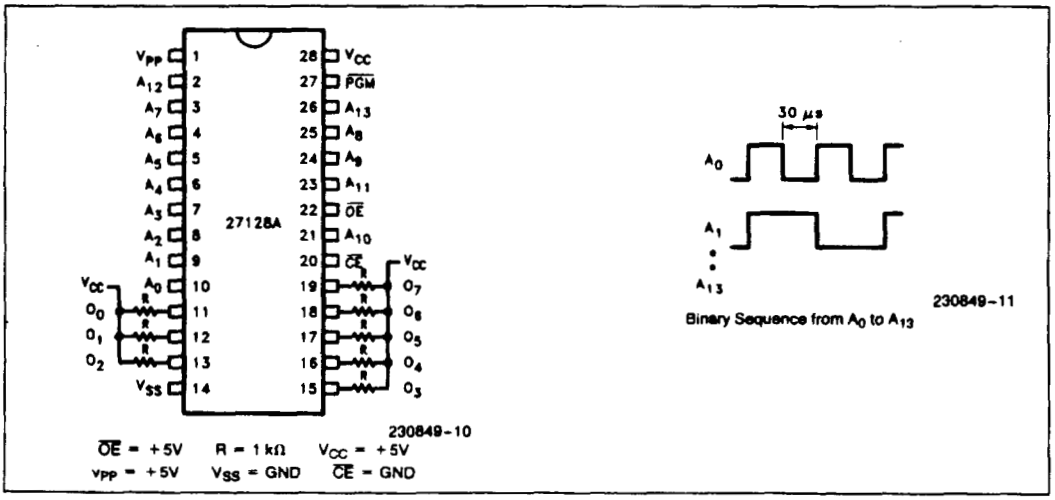
READ OPERATION

D.C. CHARACTERISTICS

Electrical Parameters of Express EPROM Products are identical to standard EPROM parameters except for:

Symbol	Parameter	TD27128A, LD27128A		Test Conditions
		Min	Max	
I_{SB}	V_{CC} Standby Current (mA)		50	$\overline{CE} = V_{IH}$, $\overline{OE} = V_{IL}$
I_{CC1}(1)	V_{CC} Active Current (mA)		125	$\overline{OE} = \overline{CE} = V_{IL}$
	V_{CC} Active Current at High Temperature (mA)		100	$\overline{OE} = \overline{CE} = V_{IL}$, $V_{PP} = V_{CC}$ $T_{Ambient} = 85°C$

NOTE:
1. The maximum current value is with Outputs O_0 to O_7 unloaded.



Burn-In Bias and Timing Diagrams

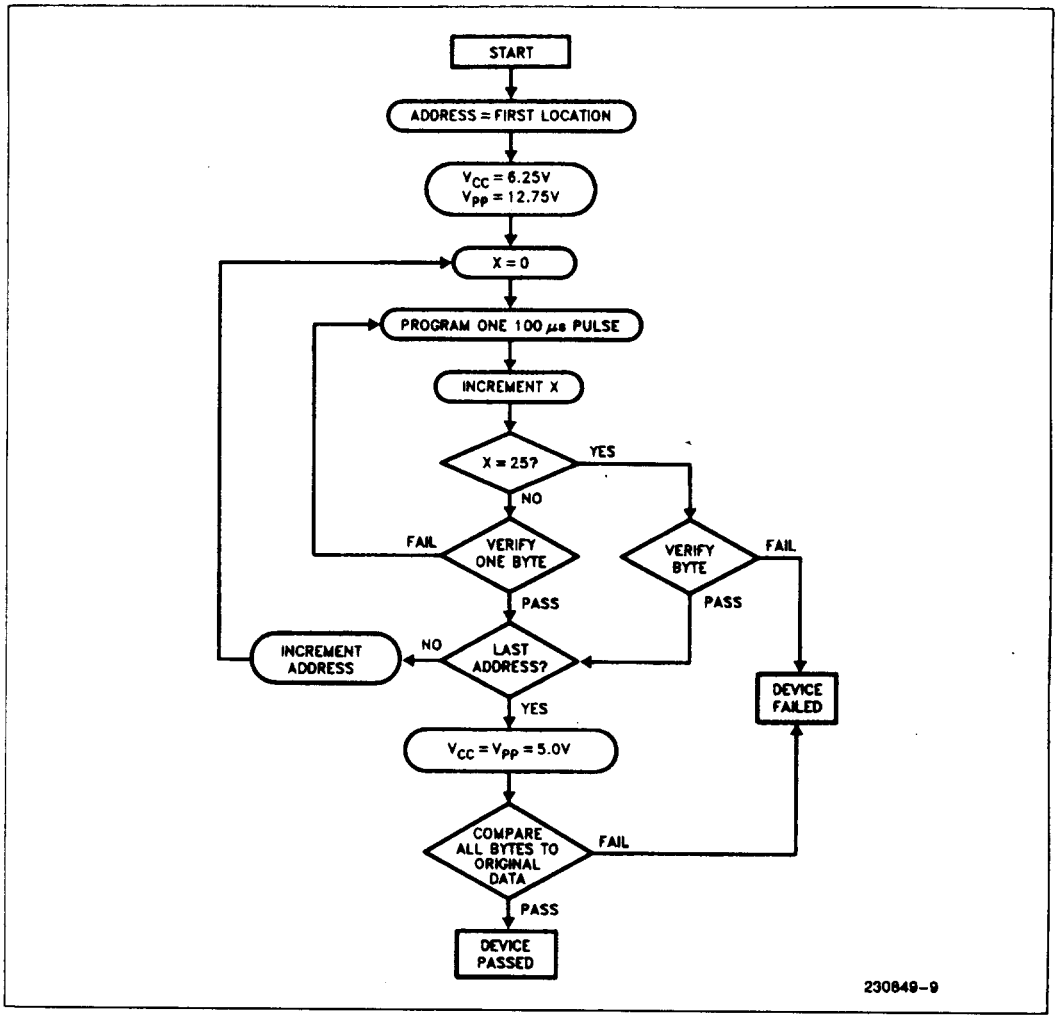


Figure 4. Quick-Pulse Programming™ Algorithm

Quick-Pulse Programming™ Algorithm (For Plastic EPROMs)

Intel's Plastic EPROMs can now be programmed using the Quick-Pulse Programming Algorithm, developed by Intel to substantially reduce the throughput time in the production programming environment. This algorithm allows Plastic devices to be programmed in under two seconds, almost a hundred fold improvement over previous algorithms. Actual programming time is a function of the PROM programmer being used.

The Quick-Pulse Programming Algorithm uses initial pulses of 100 microseconds followed by a byte verification to determine when the address byte has been successfully programmed. Up to 25 100 µs pulses per byte are provided before a failure is recognized. A flow chart of the Quick-Pulse Programming Algorithm is shown in Figure 4.

For the Quick-Pulse Programming Algorithm, the entire sequence of programming pulses and byte verifications is performed at V_{CC} = 6.25V and V_{PP} at 12.75V. When programming of the EPROM has been completed, all bytes should be compared to the original data with V_{CC} = V_{PP} = 5.0V.

In addition to the Quick-Pulse Programming Algorithm, Plastic EPROMs are also compatible with Intel's int$_e$ligent Programming Algorithm.

27C128
128K (16K x 8) CHMOS PRODUCTION AND UV ERASABLE PROMs

- **CHMOS Microcontroller and Microprocessor Compatible**
- **Low Power Consumption**
  — 100 μA Maximum Standby Current
- **Maximum Latch-Up Immunity Through EPI Processing**
  — ± 1V Input Protection
  — 14V V_{PP} Protection
- **High Performance**
  — 150 ns Access Time
- **Quick-Pulse Programming™ Algorithm Allows Rapid, Automated Programming**
  — 2 Second Throughput
- **Available in 28-Pin Cerdip and 32-Lead PLCC Packages**
  (See Packaging Spec. Order #231369)

Intel's 27C128 CHMOS EPROM is a 128K bit 5V-only memory, organized as 16,384 words of 8 bits each. The 27C128 is ideal for systems requiring low power, high performance, and noise immunity due to its CHMOS*II-E processing, and it is pin compatible with the standard Intel 27128A.

The 27C128 is offered in Ceramic DIP and Plastic Leaded Chip Carrier (PLCC) Packages. Cerdip packages provide flexibility in prototyping and R & D environments while the PLCC package is most cost effective in production environments. The Quick-Pulse Programming™ Algorithm improves programming speed by as much as one hundred times over older algorithms, further reducing costs for system manufacturers.

Intel's unique EPI processing provides excellent latch-up immunity. Prevention of latch-up is guaranteed for stresses up to 100 mA on address and data pins from $-1V$ to $V_{CC} + 1V$ and for V_{PP} voltage overshoot up to 14V.

*HMOS and CHMOS are patented processes of Intel Corporation.

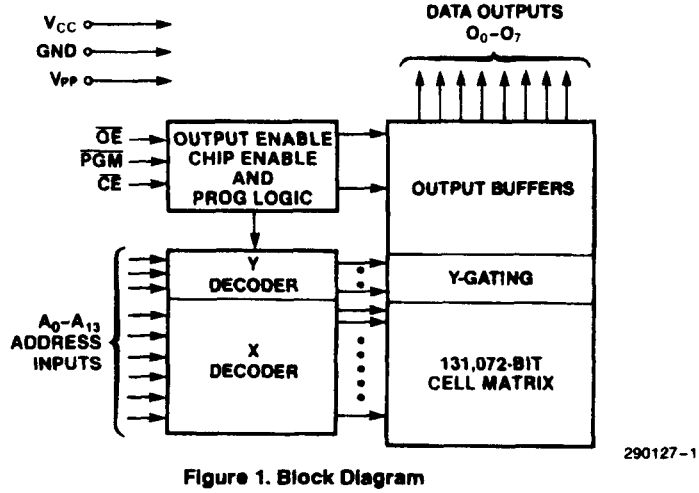


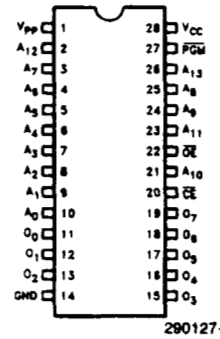
Figure 1. Block Diagram

Pin Names

A_0–A_{13}	ADDRESSES
$\overline{CE}$	CHIP ENABLE
$\overline{OE}$	OUTPUT ENABLE
O_0–O_7	OUTPUTS
$\overline{PGM}$	PROGRAM
N.C.	No Internal Connect
D.U.	Don't Use

27512 27C512	27256 27C256	2764A 27C64 87C64	2732A	2716
A_{15}	V_{PP}	V_{PP}		
A_{12}	A_{12}	A_{12}		
A_7	A_7	A_7	A_7	A_7
A_6	A_6	A_6	A_6	A_6
A_5	A_5	A_5	A_5	A_5
A_4	A_4	A_4	A_4	A_4
A_3	A_3	A_3	A_3	A_3
A_2	A_2	A_2	A_2	A_2
A_1	A_1	A_1	A_1	A_1
A_0	A_0	A_0	A_0	A_0
O_0	O_0	O_0	O_0	O_0
O_1	O_1	O_1	O_1	O_1
O_2	O_2	O_2	O_2	O_2
GND	GND	GND	GND	GND

27C128

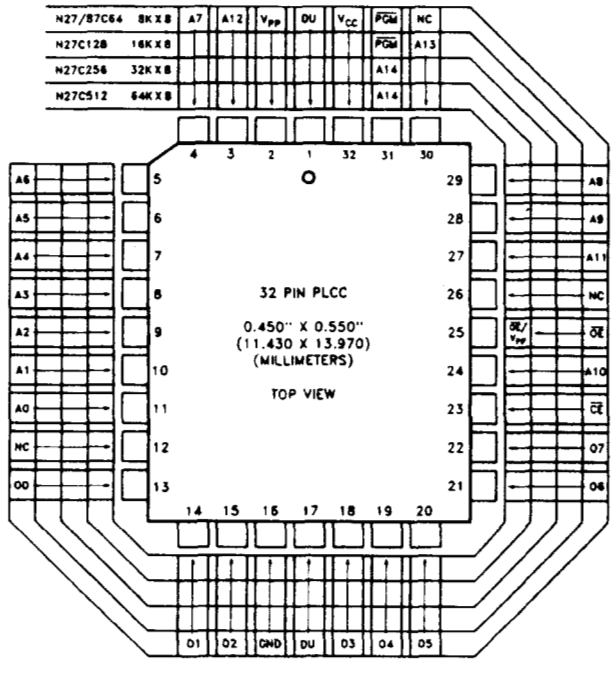


290127-2

2716	2732A	2764A 27C64 87C64	27256 27C256	27512 27C512
		V_{CC}	V_{CC}	V_{CC}
		$\overline{PGM}$	A_{14}	A_{14}
V_{CC}	V_{CC}	N.C.	A_{13}	A_{13}
A_8	A_8	A_8	A_8	A_8
A_9	A_9	A_9	A_9	A_9
V_{PP}	A_{11}	A_{11}	A_{11}	A_{11}
$\overline{OE}$	$\overline{OE}/V_{PP}$	$\overline{OE}$	$\overline{OE}$	$\overline{OE}/V_{PP}$
A_{10}	A_{10}	A_{10}	A_{10}	A_{10}
$\overline{CE}$	$\overline{CE}$	$\overline{CE}$	$\overline{CE}$	$\overline{CE}$
O_7	O_7	O_7	O_7	O_7
O_6	O_6	O_6	O_6	O_6
O_5	O_5	O_5	O_5	O_5
O_4	O_4	O_4	O_4	O_4
O_3	O_3	O_3	O_3	O_3

NOTE:
Intel "Universal Site"-Compatible EPROM Pin Configurations are Shown in the Blocks Adjacent to the 27C128 Pins.

Figure 2. Cerdip(D) Pin Configurations



290127-3

Figure 3. PLCC(N) Lead Configuration

READ MODE

The 27C128 has two control functions, both of which must be logically active in order to obtain data at the outputs. Chip Enable ($\overline{CE}$) is the power control and should be used for device selection. Output Enable ($\overline{OE}$) is the output control and should be used to gate data from the output pins, independent of device selection. Assuming that addresses are stable, the address access time (t_{ACC}) is equal to the delay from $\overline{CE}$ to output (t_{CE}). Data is available at the outputs after the delay of t_{OE} from the falling edge of $\overline{OE}$, assuming that $\overline{CE}$ has been low and addresses have been stable for at least t_{ACC}-t_{OE}.

STANDBY MODE

EPROMs can be placed in standby mode which reduces the maximum current of the device by applying a TTL-high signal to the $\overline{CE}$ input. When in standby mode, the outputs are in a high impedance state, independent of the $\overline{OE}$ input.

Two Line Output Control

Because EPROMs are usually used in larger memory arrays, Intel has provided 2 control lines which accommodate this multiple memory connection. The two control lines allow for:

a) the lowest possible memory power dissipation, and

b) complete assurance that output bus contention will not occur.

To use these two control lines most efficiently, $\overline{CE}$ should be decoded and used as the primary device selecting function, while $\overline{OE}$ should be made a common connection to all devices in the array and connected to the $\overline{READ}$ line from the system control bus. This assures that all deselected memory devices are in their low power standby mode and that the output pins are active only when data is desired from a particular memory device.

SYSTEM CONSIDERATIONS

The power switching characteristics of EPROMs require careful decoupling of the devices. The supply current, I_{CC}, has three segments that are of interest to the system designer—the standby current level, the active current level, and the transient current peaks that are produced by the falling and rising edges of Chip Enable. The magnitude of these transient and inductive current peaks is dependent on the output capacitive and inductive loading of the device. The associated transient voltage peaks can be suppressed by complying with Intel's Two-Line Control, and by properly selected decoupling capacitors. It is recommended that a 0.1 μF ceramic capacitor be used on every device between V_{CC} and GND. This should be a high frequency capacitor for low inherent inductance and should be placed as close to the device as possible. In addition, a 4.7 μF bulk electrolytic capacitor should be used between V_{CC} and GND for every eight devices. The bulk capacitor should be located near where the power supply is connected to the array. The purpose of the bulk capacitor is to overcome the voltage droop caused by the inductive effect of PC board-traces.

PROGRAMMING MODES

Caution: Exceeding 14V on V_{PP} will permanently damage the device.

Initially, and after each erasure, all bits of the EPROM are in the "1" state. Data is introduced by selectively programming "0s" into the desired bit locations. Although only "0s" will be programmed, both "1s" and "0s" can be present in the data word. The only way to change a "0" to a "1" is by ultraviolet light erasure.

The device is in the programming mode when V_{PP} is raised to its programming voltage (See Table 2) and $\overline{CE}$ and $\overline{PGM}$ are both at TTL low and $\overline{OE}$ = V_{IH}. The data to be programmed is applied 8 bits in parallel to the data output pins. The levels required for the address and data inputs are TTL.

Program Inhibit

Programming of multiple EPROMs in parallel with different data is easily accomplished by using the Program Inhibit mode. A high-level $\overline{CE}$ or $\overline{PGM}$ input inhibits the other devices from being programmed. Except for $\overline{CE}$, all like inputs (including $\overline{OE}$) of the parallel EPROMs may be common. A TTL low-level pulse applied to the $\overline{PGM}$ input with V_{PP} at its programming voltage and $\overline{CE}$ = V_{IL} will program the selected device.

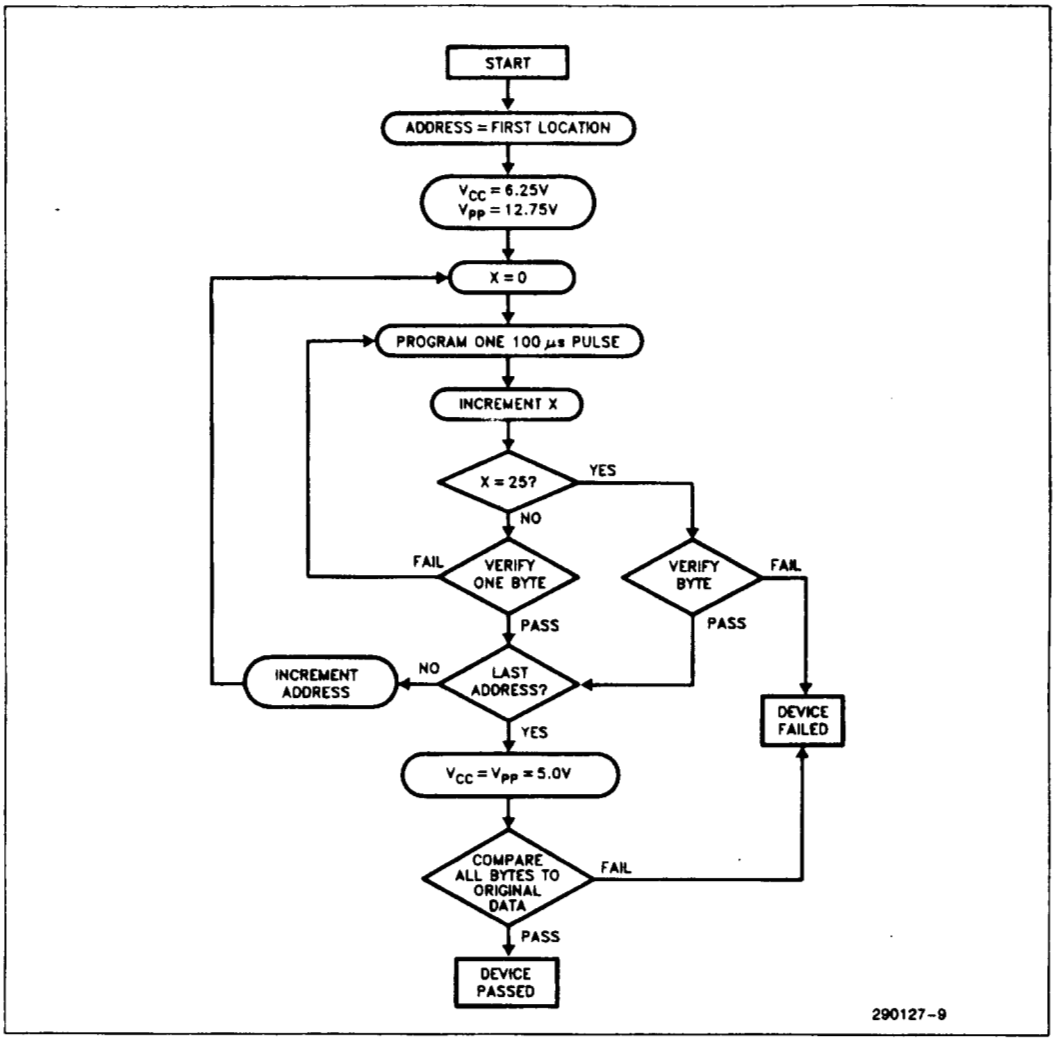


Figure 5. Quick-Pulse Programming™ Algorithm

Quick-Pulse Programming™ Algorithm

Intel's 27C128 EPROM is programmed using the Quick-Pulse Programming Algorithm, developed by Intel to substantially reduce the throughput time in the production environment. This algorithm allows the device to be programmed in under two seconds, almost a hundred fold improvement over previous algorithms. Actual programming time is a function of the PROM programmer being used.

The Quick-Pulse Programming Algorithm uses initial pulses of 100 microseconds followed by a byte verification to determine when the address byte has been successfully programmed. Up to 25 100 µs pulses per byte are provided before a failure is recognized. A flowchart of the Quick-Pulse Programming Algorithm is shown in Figure 5.

For the Quick Pulse Programming Algorithm, the entire sequence of programming pulses and byte verifications is performed at V_{CC} = 6.25V and V_{PP} at 12.75V. When programming of the EPROM has been completed, all bytes should be compared to the original data with V_{CC} = V_{PP} = 5.0V.

27256
256K (32K x 8) PRODUCTION AND UV ERASABLE PROMS

- **New Quick-Pulse Programming™ Algorithm for Plastic P27256**
  - **— 4 Second Programming**
  - **— int$_e$ligent Programming™ Algorithm Compatible**
- **Fast Access Time**
  - **— 170 ns D27256-1**
  - **— 200 ns P27256-2**
- **int$_e$ligent Identifier™ Mode**
- **Plastic Production P27256 is Compatible with Auto-Insertion Equipment**
- **Moisture Resistant**
- **Industry Standard Pinout . . . JEDEC Approved . . . 28 Lead Cerdip and Plastic Package**

(See Packaging Spec, Order #231369)

The Intel 27256 is a 5V only, 262,144-bit Ultraviolet Erasable (Cerdip)/plastic production (P27256) electrically programmable read-only memory (EPROM). Organized as 32K words by 8 bits, individual bytes can be accessed in less than 170 ns (27256-1). This is compatible with high performance microprocessors, such as the Intel iAPX 186, allowing full speed operation without the addition of performance-degrading WAIT states. The 27256 is also directly compatible with Intel's 8051 family of microcontrollers.

The Plastic P27256 is ideal for high volume production environments where code flexibility is crucial. Plastic packaging is also well-suited to auto-insertion equipment in cost-effective automated assembly lines. Intel's new Quick-Pulse Programming Algorithm enables the P27256 to be programmed within four seconds (plus programmer overhead). Programming equipment which takes advantage of this innovation will electronically identify the EPROM with the help of the int$_e$ligent Identifier and rapidly program it using a superior programming method. The int$_e$ligent Programming Algorithm may be utilized in the absence of such equipment.

The 27256 enables implementation of new, advanced systems with firmware-intensive architectures. The combination of the 27256's high-density, cost-effective EPROM storage, and new advanced microprocessors having megabit addressing capability provides designers with opportunities to engineer user-friendly, high reliability, high-performance systems.

The 27256's large storage capability of 32 K-bytes enables it to function as a high-density software carrier. Entire operating systems, diagnostics, high-level language programs and specialized application software can reside in a 27256 EPROM directly on a system's memory bus. This permits immediate microprocessor access and execution of software and elminates the need for time-consuming disk accesses and downloads.

Two-line control and JEDEC-approved, 28-pin packaging are standard features of all Intel high-density EPROMs. This assures easy microprocessor interfacing and minimum design efforts when upgrading, adding, or choosing between nonvolatile memory alternatives.

The 27256 is manufactured using Intel's advanced HMOS*II-E technology.

*HMOS is a patented process of Intel Corporation.

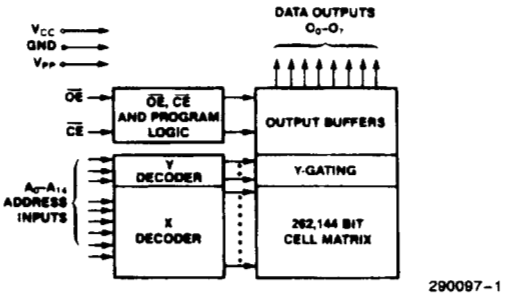


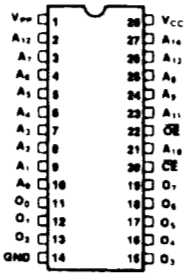
Figure 1. Block Diagram

Pin Names

A_0–A_{14}	Addresses
$\overline{CE}$	Chip Enable
$\overline{OE}$	Output Enable
O_0–O_7	Outputs
N.C.	No Connect

27512 27C512	27128A 27C128	2764A 27C64 87C64	2732A	2716
A_{15}	V_{PP}	V_{PP}		
A_{12}	A_{12}	A_{12}		
A_7	A_7	A_7	A_7	A_7
A_6	A_6	A_6	A_6	A_6
A_5	A_5	A_5	A_5	A_5
A_4	A_4	A_4	A_4	A_4
A_3	A_3	A_3	A_3	A_3
A_2	A_2	A_2	A_2	A_2
A_1	A_1	A_1	A_1	A_1
A_0	A_0	A_0	A_0	A_0
O_0	O_0	O_0	O_0	O_0
O_1	O_1	O_1	O_1	O_1
O_2	O_2	O_2	O_2	O_2
GND	GND	GND	GND	GND

27256
P27256



290097-2

2716	2732A	2764A 27C64 87C64	27182A 27C128	27512 27C512
		V_{CC} / $\overline{PGM}$	V_{CC} / $\overline{PGM}$	V_{CC}
V_{CC}	V_{CC}	N.C.	A_{13}	A_{13}
A_8	A_8	A_8	A_8	A_8
A_9	A_9	A_9	A_9	A_9
V_{PP}	A_{11}	A_{11}	A_{11}	A_{11}
$\overline{OE}$	$\overline{OE}/V_{PP}$	$\overline{OE}$	$\overline{OE}$	$\overline{OE}/V_{PP}$
A_{10}	A_{10}	A_{10}	A_{10}	A_{10}
$\overline{CE}$	$\overline{CE}$	$\overline{CE}$	$\overline{CE}$	$\overline{CE}$
O_7	O_7	O_7	O_7	O_7
O_6	O_6	O_6	O_6	O_6
O_5	O_5	O_5	O_5	O_5
O_4	O_4	O_4	O_4	O_4
O_3	O_3	O_3	O_3	O_3

NOTE:
Intel "Universal Site"-Compatible EPROM pin configurations are shown in the blocks adjacent to the P27256 pins.

Figure 2. Cerdip/Plastic DIP Pin Configuration

EXTENDED TEMPERATURE (EXPRESS) EPROMs

The Intel EXPRESS EPROM family is a series of electrically programmable read only memories which have received additional processing to enhance product characteristics. EXPRESS processing is available for several densities of EPROM, allowing the choice of appropriate memory size to match system applications. EXPRESS EPROM products are available with 168 ±8 hour, 125°C dynamic burn-in using Intel's standard bias configuration. This process exceeds or meets most industry specifications of burn-in. The standard EXPRESS EPROM operating temperature range is 0°C to 70°C. Extended operating temperature range (−40°C to +85°C) EXPRESS products are available. Like all Intel EPROMs, the EXPRESS EPROM family is inspected to 0.1% electrical AQL. This may allow the user to reduce or eliminate incoming inspection testing.

EXPRESS EPROM PRODUCT FAMILY

PRODUCT DEFINITIONS

Type	Operating Temperature	Burn-In 125°C (hr)
Q	0°C to +70°C	168 ±8
T	−40°C to +85°C	None
L	−40°C to +85°C	168 ±8

EXPRESS OPTIONS

27256 VERSIONS

Packaging Options	
Speed Versions	Cerdip
-20	Q, T, L

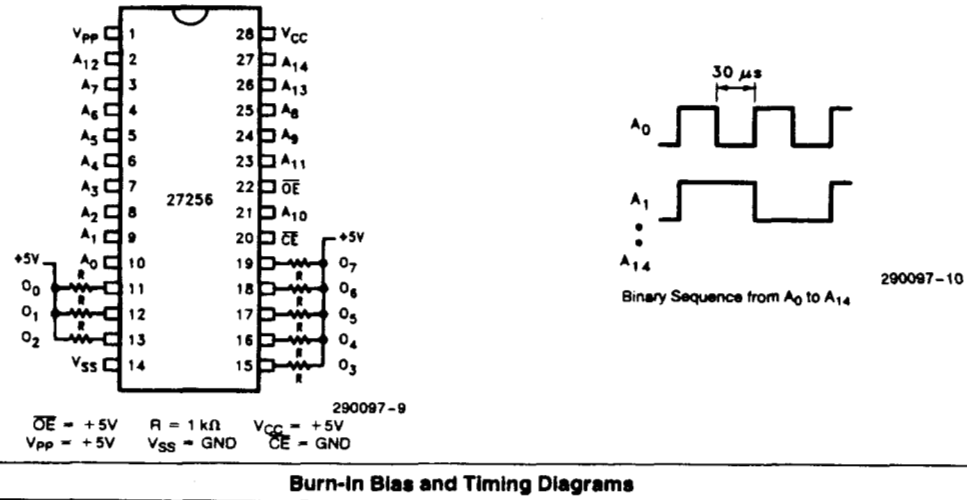
READ OPERATION

D.C. CHARACTERISTICS

Electrical parameters of EXPRESS EPROM products are identical to standard EPROM parameters except for:

Symbol	Parameter	TD27256 LD27256		Test Conditions
		Min	Max	
I_{SB}	V_{CC} Standby Current (mA)		50	$\overline{CE} = V_{IH}$, $\overline{OE} = V_{IL}$
I_{CC}(1)	V_{CC} Active Current (mA)		125	$\overline{OE} = \overline{CE} = V_{IL}$

NOTE:
1. The maximum current value is with outputs O_0 to O_7 unloaded.



Burn-In Bias and Timing Diagrams

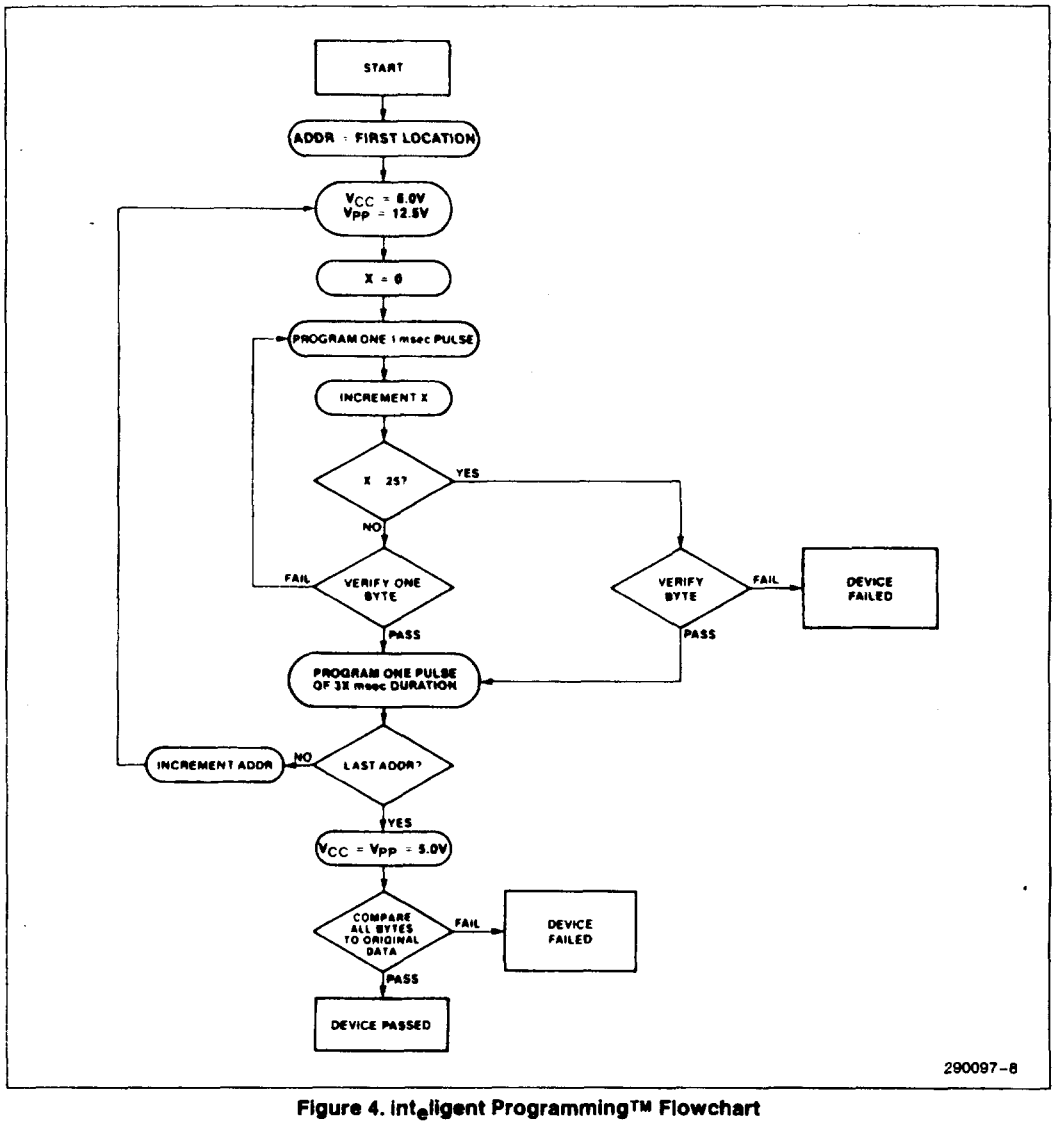


Figure 4. int$_e$ligent Programming™ Flowchart

int$_e$ligent Programming™ Algorithm

The int$_e$ligent Programming Algorithm has been a standard in the industry for the past few years. A flowchart of the int$_e$ligent Programming Algorithm is shown in Figure 4.

The int$_e$ligent Programming Algorithm utilizes two different pulse types: initial and overprogram. The duration of the initial $\overline{CE}$ pulse(s) is one millisecond, which will then be followed by a longer overprogram pulse of length 3X msec. X is an iteration counter and is equal to the number of the initial one millisecond pulses applied to a particular location, before a correct verify occurs. Up to 25 one-millisecond pulses per byte are provided for before the overprogram pulse is applied.

The entire sequence of program pulses and byte verifications is performed at V_{CC} = 6.0V and V_{PP} = 12.5V. When the int$_e$ligent Programming cycle has been completed, all bytes should be compared to the original data with $V_{CC} = V_{PP} = 5.0V$.

27512
512K (64K x 8) PRODUCTION AND UV ERASABLE PROM

- **Software Carrier Capability**
- **170 ns Maximum Access Time**
- **Two-Line Control**
- **int$_e$ligent Identifier™ Mode**
  — Automated Programming Operations
- **TTL Compatible**
- **Low Power**
  — 125 mA max. Active
  — 40 mA max. Standby
- **int$_e$ligent Programming™ Algorithm**
- **Available in 28-Pin Cerdip**
  (See packaging spec order #231369)

The Intel 27512 is a 5V-only, 524,288-bit ultraviolet Erasable and Electrically Programmable Read Only Memory (EPROM) organized as 64K words by 8 bits. This ensures compatibility with high-performance microprocessors, such as the Intel 8 MHz iAPX 286, allowing full speed operation without the addition of performance-degrading WAIT states. The 27512 is also directly compatible with Intel's 8051 family of microcontrollers.

The 27512 enables implementation of new, advanced systems with firmware intensive architectures. The combination of the 27512's high-density, cost-effective EPROM storage, and new advanced microprocessors having megabyte addressing capability provides designers with opportunities to engineer user-friendly, high-reliability, high-performance systems.

The 27512's large storage capability of 64 K-bytes enables it to function as a high-density software carrier. Entire operating systems, diagnostics, high-level language programs and specialized application software can reside in a 27512 EPROM directly on a system's memory bus. This permits immediate microprocessor access and execution of software and eliminates the need for time-consuming disk accesses and downloads.

Two-line control and JEDEC-approved, 28-pin packaging are standard features of all Intel high-density EPROMs. This assures easy microprocessor interfacing and minimum design efforts when upgrading, adding, or choosing between nonvolatile memory alternatives.

The 27512 is manufactured using Intel's advanced HMOS *II-E technology.

*HMOS is a patented process of Intel Corporation.

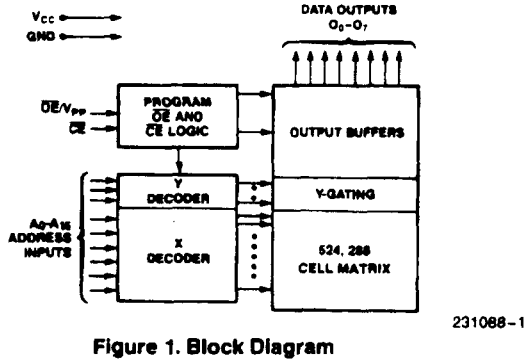
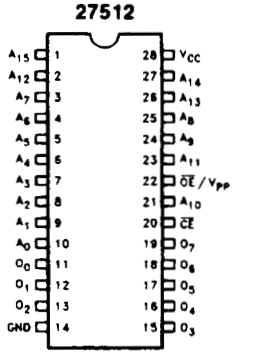


Figure 1. Block Diagram

Pin Names

A_0–A_{15}	Addresses
$\overline{CE}$	Chip Enable
$\overline{OE}/V_{PP}$	Outputs Enable/V_{PP}
O_0–O_7	Outputs
D.U.	Don't Use

27256 27C256	27128A 27C128	2764A 27C64 87C64	2732A	2716
V_{PP}	V_{PP}	V_{PP}		
A_{12}	A_{12}	A_{12}		
A_7	A_7	A_7	A_7	A_7
A_6	A_6	A_6	A_6	A_6
A_5	A_5	A_5	A_5	A_5
A_4	A_4	A_4	A_4	A_4
A_3	A_3	A_3	A_3	A_3
A_2	A_2	A_2	A_2	A_2
A_1	A_1	A_1	A_1	A_1
A_0	A_0	A_0	A_0	A_0
O_0	O_0	O_0	O_0	O_0
O_1	O_1	O_1	O_1	O_1
O_2	O_2	O_2	O_2	O_2
GND	GND	GND	GND	GND



231088-2

2716	2732A	2764A 27C64 87C64	27128 27128A	27256 27C256
		V_{CC}	V_{CC}	V_{CC}
		$\overline{PGM}$	$\overline{PGM}$	A_{14}
V_{CC}	V_{CC}	N.C.	A_{13}	A_{13}
A_8	A_8	A_8	A_8	A_8
A_9	A_9	A_9	A_9	A_9
V_{PP}	A_{11}	A_{11}	A_{11}	A_{11}
$\overline{OE}$	$\overline{OE}/V_{PP}$	$\overline{OE}$	$\overline{OE}$	$\overline{OE}$
A_{10}	A_{10}	A_{10}	A_{10}	A_{10}
$\overline{CE}$	$\overline{CE}$	$\overline{CE}$ ALE/$\overline{CE}$	$\overline{CE}$	$\overline{CE}$
O_7	O_7	O_7	O_7	O_7
O_6	O_6	O_6	O_6	O_6
O_5	O_5	O_5	O_5	O_5
O_4	O_4	O_4	O_4	O_4
O_3	O_3	O_3	O_3	O_3

Figure 2. Pin Configurations

EXTENDED TEMPERATURE (EXPRESS) EPROMs

The Intel EXPRESS EPROM family is a series of electrically programmable read only memories which have received additional processing to enhance product characteristics. EXPRESS processing is available for several densities of EPROM, allowing the choice of appropriate memory size to match system applications. EXPRESS EPROM products are available with 168 ±8 hours, 125°C dynamic burn-in using Intel's standard bias configuration. This process exceeds or meets most industry specifications of burn-in. The standard EXPRESS EPROM operating temperature range is 0°C to 70°C. Extended operating temperature range (−40°C to +85°C) EXPRESS products are available. Like all Intel EPROMs, the EXPRESS EPROM family is inspected to 0.1% electrical AQL. This may allow the user to reduce or eliminate incoming inspection testing.

EXPRESS EPROM PRODUCT FAMILY

PRODUCT DEFINITIONS

Type	Operating Temperature	Burn-in 125°C (hr)
Q	0°C to +70°C	168 ±8
T	−40°C to +85°C	None
L	−40°C to +85°C	168 ±8

EXPRESS OPTIONS

27512 VERSIONS

Packaging Options	
Speed Versions	**Cerdip**
-2	Q
-STD, -25, -30	Q, T, L
-3	L

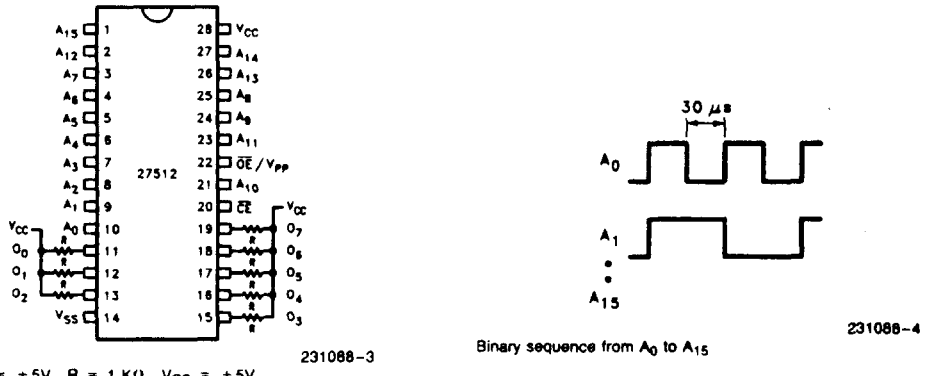
READ OPERATION

D.C. CHARACTERISTICS

Electrical parameters of EXPRESS EPROM products are identical to standard EPROM parameters except for:

Symbol	Parameter	TD27512 LD27512		Test Conditions
		Min	Max	
I_{SB}	V_{CC} Standby Current (mA)		50	$\overline{CE} = V_{IH}$, $\overline{OE}/V_{PP} = V_{IL}$
I_{CC1}(1)	V_{CC} Active Current (mA)		150	$\overline{OE}/V_{PP} = \overline{CE} = V_{IL}$
	V_{CC} Active Current at High Temperature (mA)		125	$\overline{OE}/V_{PP} = \overline{CE} = V_{IL}$ $T_{Ambient} = 85°C$

NOTE:
1. The maximum current value is with outputs O_0 to O_7 unloaded.



Burn-In Bias and Timing Diagrams

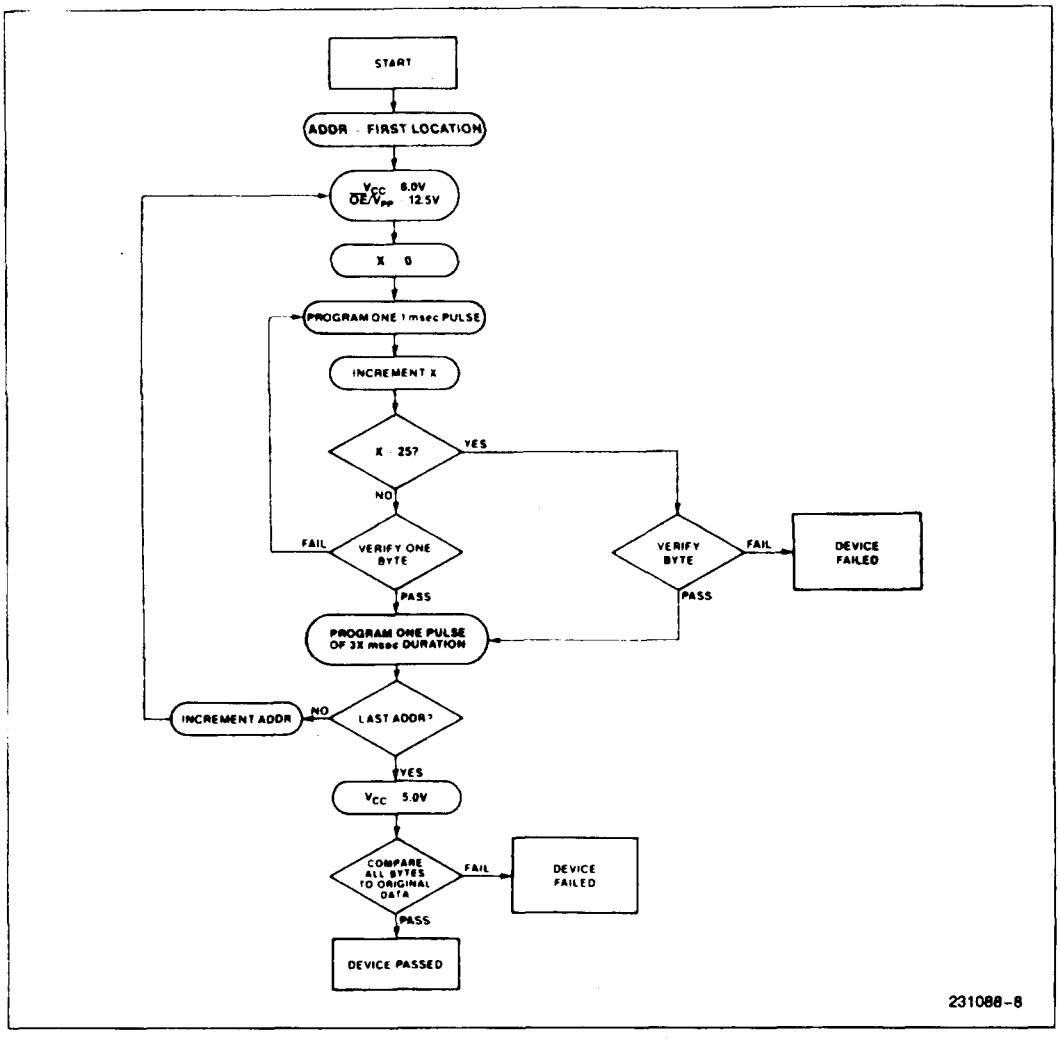


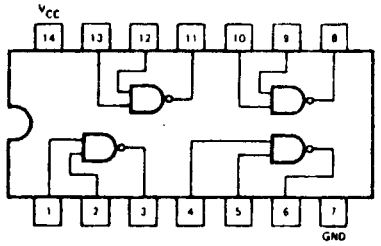
Figure 5. int$_e$ligent Programming™ Flowchart

int$_e$ligent Programming™ Algorithm

The int$_e$ligent Programming Algorithm programs Intel EPROMs using an efficient and reliable method particularly suited to the production programming environment. Typical programming time for individual devices are on the order of six minutes. Actual programming times may vary due to differences in programming equipment. Programming reliability is also ensured as the incremental program margin of each byte is continually monitored to determine when it has been successfully programmed. A flowchart of the int$_e$ligent Programming Algorithm is shown in Figure 4.

The int$_e$ligent Programming Algorithm utilizes two different pulse types: initial and overprogram. The duration of the initial pulse(s) is one millisecond, which will then be followed by a longer overprogram pulse of length 3X msec. X is an iteration counter and is equal to the number of the initial one millisecond pulses applied to a particular location, before a correct verify occurs. Up to 25 one-millisecond pulses per byte are provided for before the overprogram pulse is applied. ***The entire sequence of program pulses and byte verifications is performed at V_{CC} = 6.0V.*** When the int$_e$ligent Programming cycle has been completed, all bytes should be compared to the original data with V_{CC} = 5.0V.

QUAD 2-INPUT NAND GATE



GUARANTEED OPERATING RANGES

PART NUMBERS	SUPPLY VOLTAGE			TEMPERATURE
	MIN	TYP	MAX	
T54LS00X	4.5 V	5.0 V	5.5 V	-55°C to 125°C
T74LS00X	4.75 V	5.0 V	5.25 V	0°C to +70°C

X = package type: D for Ceramic Dip, B for Plastic Dip. See Packaging Information Section for packages available on this product

DC CHARACTERISTICS OVER OPERATING TEMPERATURE RANGE (unless otherwise specified)

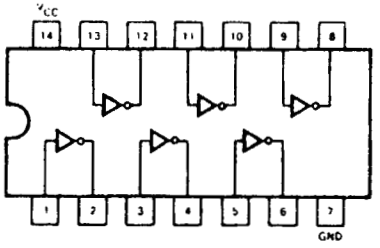
SYMBOL	PARAMETER		LIMITS MIN	LIMITS TYP	LIMITS MAX	UNITS	TEST CONDITIONS (Note 1)
V_{IH}	Input HIGH Voltage		2.0			V	Guaranteed Input HIGH Voltage
V_{IL}	Input LOW Voltage	54			0.7	V	Guaranteed Input LOW Voltage
		74			0.8		
V_{CD}	Input Clamp Diode Voltage			-0.65	-1.5	V	V_{CC} = MIN, I_{IN} = -18 mA
V_{OH}	Output HIGH Voltage	54	2.5	3.4		V	V_{CC} = MIN, I_{OH} = -400 µA, V_{IN} = V_{IL}
		74	2.7	3.4			
V_{OL}	Output LOW Voltage	54,74		0.25	0.4	V	V_{CC} = MIN, I_{OL} = 4.0 mA, V_{IN} = 2.0 V
		74		0.35	0.5	V	V_{CC} = MIN, I_{OL} = 8.0 mA, V_{IN} = 2.0 V
I_{IH}	Input HIGH Current			1.0	20	µA	V_{CC} = MAX, V_{IN} = 2.7 V
					0.1	mA	V_{CC} = MAX, V_{IN} = 10 V
I_{IL}	Input LOW Current				-0.36	mA	V_{CC} = MAX, V_{IN} = 0.4 V
I_{OS}	Output Short Circuit Current (Note 3)		-20		-100	mA	V_{CC} = MAX, V_{OUT} = 0 V
I_{CCH}	Supply Current HIGH			0.8	1.6	mA	V_{CC} = MAX, V_{IN} = 0 V
I_{CCL}	Supply Current LOW			2.4	4.4	mA	V_{CC} = MAX, Inputs Open

AC CHARACTERISTICS: T_A = 25°C (See Page 273 for Waveforms)

SYMBOL	PARAMETER	LIMITS MIN	LIMITS TYP	LIMITS MAX	UNITS	TEST CONDITIONS
t_{PLH}	Turn Off Delay, Input to Output	3.0	5.0	10	ns	V_{CC} = 5.0 V
t_{PHL}	Turn On Delay, Input to Output	3.0	5.0	10	ns	C_L = 15 pF

NOTES:
1. For conditions shown as MIN or MAX, use the appropriate value specified under recommended operating conditions for the applicable device type.
2. Typical limits are at V_{CC} = 5.0 V, T_A = 25°C.
3. Not more than one output should be shorted at a time.

HEX INVERTER



GUARANTEED OPERATING RANGES

PART NUMBERS	SUPPLY VOLTAGE			TEMPERATURE
	MIN	TYP	MAX	
T54LS04X	4.5 V	5.0 V	5.5 V	−55°C to 125°C
T74LS04X	4.75 V	5.0 V	5.25 V	0°C to +70°C

X = package type; D for Ceramic Dip, B for Plastic Dip. See Packaging Information Section for packages available on this product.

DC CHARACTERISTICS OVER OPERATING TEMPERATURE RANGE (unless otherwise specified)

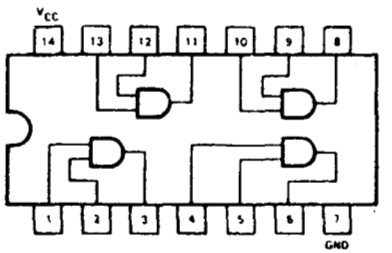
SYMBOL	PARAMETER		LIMITS MIN	LIMITS TYP	LIMITS MAX	UNITS	TEST CONDITIONS (Note 1)
V_{IH}	Input HIGH Voltage		2.0			V	Guaranteed Input HIGH Voltage
V_{IL}	Input LOW Voltage	54			0.7	V	Guaranteed Input LOW Voltage
		74			0.8		
V_{CD}	Input Clamp Diode Voltage			−0.65	−1.5	V	V_{CC} = MIN, I_{IN} = −18 mA
V_{OH}	Output HIGH Voltage	54	2.5	3.4		V	V_{CC} = MIN, I_{OH} = −400 μA, V_{IN} = V_{IL}
		74	2.7	3.4			
V_{OL}	Output LOW Voltage	54,74		0.25	0.4	V	V_{CC} = MIN, I_{OL} = 4.0 mA, V_{IN} = 2.0 V
		74		0.35	0.5	V	V_{CC} = MIN, I_{OL} = 8.0 mA, V_{IN} = 2.0 V
I_{IH}	Input HIGH Current			1.0	20	μA	V_{CC} = MAX, V_{IN} = 2.7 V
					0.1	mA	V_{CC} = MAX, V_{IN} = 10 V
I_{IL}	Input LOW Current				−0.36	mA	V_{CC} = MAX, V_{IN} = 0.4 V
I_{OS}	Output Short Circuit Current (Note 3)		−20		100	mA	V_{CC} = MAX, V_{OUT} = 0 V
I_{CCH}	Supply Current HIGH			1.2	2.4	mA	V_{CC} = MAX, V_{IN} = 0 V
I_{CCL}	Supply Current LOW			3.6	6.6	mA	V_{CC} = MAX, Inputs Open

AC CHARACTERISTICS: T_A = 25°C (See Page 273 for Waveforms)

SYMBOL	PARAMETER	LIMITS MIN	LIMITS TYP	LIMITS MAX	UNITS	TEST CONDITIONS
t_{PLH}	Turn Off Delay, Input to Output	3.0	5.0	10	ns	V_{CC} = 5.0 V
t_{PHL}	Turn On Delay, Input to Output	3.0	5.0	10	ns	C_L = 15 pF

NOTES
1. For conditions shown as MIN or MAX, use the appropriate value specified under recommended operating conditions for the applicable device type
2. Typical limits are at V_{CC} = 5.0 V, T_A = 25°C
3. Not more than one output should be shorted at a time

QUAD 2-INPUT AND GATE



GUARANTEED OPERATING RANGES

PART NUMBERS	SUPPLY VOLTAGE			TEMPERATURE
	MIN	TYP	MAX	
T54LS08X	4.5 V	5.0 V	5.5 V	-55°C to 125°C
T74LS08X	4.75 V	5.0 V	5.25 V	0°C to +70°C

X = package type: D for Ceramic Dip, B for Plastic Dip. See Packaging Information Section for packages available on this product.

DC CHARACTERISTICS OVER OPERATING TEMPERATURE RANGE (unless otherwise specified)

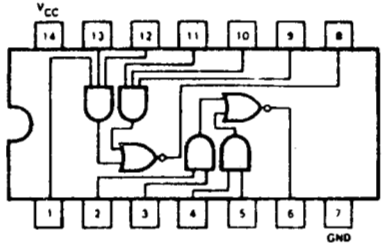
SYMBOL	PARAMETER		LIMITS			UNITS	TEST CONDITIONS (Note 1)
			MIN	TYP	MAX		
V_{IH}	Input HIGH Voltage		2.0			V	Guaranteed Input HIGH Voltage
V_{IL}	Input LOW Voltage	54			0.7	V	Guaranteed Input LOW Voltage
		74			0.8		
V_{CD}	Input Clamp Diode Voltage			-0.65	-1.5	V	V_{CC} = MIN, I_{IN} = -18 mA
V_{OH}	Output HIGH Voltage	54	2.5	3.4		V	V_{CC} = MIN, I_{OH} = -400 µA, V_{IN} = V_{IH}
		74	2.7	3.4			
V_{OL}	Output LOW Voltage	54, 74		0.25	0.4	V	V_{CC} = MIN, I_{OL} = 4.0 mA, V_{IN} = V_{IL}
		74		0.35	0.5	V	V_{CC} = MIN, I_{OL} = 8.0 mA, V_{IN} = V_{IL}
I_{IH}	Input HIGH Current			1.0	20	µA	V_{CC} = MAX, V_{IN} = 2.7 V
					0.1	mA	V_{CC} = MAX, V_{IN} = 10 V
I_{IL}	Input LOW Current				-0.36	mA	V_{CC} = MAX, V_{IN} = 0.4 V
I_{OS}	Output Short Circuit Current (Note 3)		-20		-100	mA	V_{CC} = MAX, V_{OUT} = 0 V
I_{CCH}	Supply Current HIGH			2.4	4.8	mA	V_{CC} = MAX, Inputs Open
I_{CCL}	Supply Current LOW			4.4	8.8	mA	V_{CC} = MAX, V_{IN} = 0 V

AC CHARACTERISTICS: T_A = 25°C (See Page 273 for Waveforms)

SYMBOL	PARAMETER	LIMITS			UNITS	TEST CONDITIONS
		MIN	TYP	MAX		
t_{PLH}	Turn Off Delay, Input to Output		8.0	13	ns	V_{CC} = 5.0 V
t_{PHL}	Turn On Delay, Input to Output		7.5	11	ns	C_L = 15 pF

NOTES
1. For conditions shown as MIN or MAX, use the appropriate value specified under recommended operating conditions for the applicable device type.
2. Typical limits are at V_{CC} = 5.0 V, T_A = 25°C.
3. Not more than one output should be shorted at a time.

DUAL 2-WIDE 2-INPUT/3-INPUT AND-OR-INVERT GATE



GUARANTEED OPERATING RANGES

PART NUMBERS	SUPPLY VOLTAGE			TEMPERATURE
	MIN	TYP	MAX	
T54LS51X	4.5 V	5.0 V	5.5 V	−55°C to 125°C
T74LS51X	4.75 V	5.0 V	5.25 V	0°C to +70°C

X = package type; D for Ceramic Dip, B for Plastic Dip. See Packaging Information Section for packages available on this product

DC CHARACTERISTICS OVER OPERATING TEMPERATURE RANGE (unless otherwise specified)

SYMBOL	PARAMETER		LIMITS MIN	LIMITS TYP	LIMITS MAX	UNITS	TEST CONDITIONS (Note 1)
V_{IH}	Input HIGH Voltage		2.0			V	Guaranteed Input HIGH Voltage
V_{IL}	Input LOW Voltage	54			0.7	V	Guaranteed Input LOW Voltage
		74			0.8		
V_{CD}	Input Clamp Diode Voltage			−0.65	−1.5	V	V_{CC} = MIN, I_{IN} = −18 mA
V_{OH}	Output HIGH Voltage	54	2.5	3.4		V	V_{CC} = MIN, I_{OH} = −400 μA, V_{IN} = V_{IL}
		74	2.7	3.4			
V_{OL}	Output LOW Voltage	54,74		0.25	0.4	V	V_{CC} = MIN, I_{OL} = 4.0 mA, V_{IN} = 2.0 V
		74		0.35	0.5	V	V_{CC} = MIN, I_{OL} = 8.0 mA, V_{IN} = 2.0 V
I_{IH}	Input HIGH Current			1.0	20	μA	V_{CC} = MAX, V_{IN} = 2.7 V
					0.1	mA	V_{CC} = MAX, V_{IN} = 10 V
I_{IL}	Input LOW Current				−0.36	mA	V_{CC} = MAX, V_{IN} = 0.4 V
I_{OS}	Output Short Circuit Current (Note 3)		−20		−100	mA	V_{CC} = MAX, V_{OUT} = 0 V
I_{CCH}	Supply Current HIGH			0.8	1.6	mA	V_{CC} = MAX, V_{IN} = 0 V
I_{CCL}	Supply Current LOW			1.4	2.8	mA	V_{CC} = MAX, Inputs Open

AC CHARACTERISTICS: T_A = 25°C (See Page 273 for Waveforms)

SYMBOL	PARAMETER	LIMITS MIN	LIMITS TYP	LIMITS MAX	UNITS	TEST CONDITIONS
t_{PLH}	Turn Off Delay, Input to Output		8.0	13	ns	V_{CC} = 5.0 V
t_{PHL}	Turn On Delay, Input to Output		8.0	13	ns	C_L = 15 pF

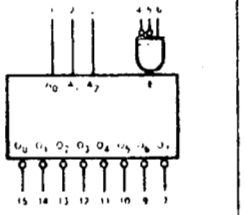
NOTES
1. For conditions shown as MIN or MAX, use the appropriate value specified under recommended operating conditions for the applicable device type.
2. Typical limits are at V_{CC} = 5.0 V, T_A = 25°C.
3. Not more than one output should be shorted at a time

T54LS138/T74LS138
1-OF-8 DECODER/DEMULTIPLEXER

DESCRIPTION – The LSTTL/MSI T54LS138/T74LS138 is a high speed 1-of-8 Decoder/Demultiplexer. This device is ideally suited for high speed bipolar memory chip select address decoding. The multiple input enables allow parallel expansion to a 1-of-24 decoder using just three LS138 devices or to a 1-of-32 decoder using four LS138s and one inverter. The LS138 is fabricated with the Schottky barrier diode process for high speed and is completely compatible with all SGS-ATES TTL families.

- **DEMULTIPLEXING CAPABILITY**
- **MULTIPLE INPUT ENABLE FOR EASY EXPANSION**
- **TYPICAL POWER DISSIPATION OF 32 mW**
- **ACTIVE LOW MUTUALLY EXCLUSIVE OUTPUTS**
- **INPUT CLAMP DIODES LIMIT HIGH SPEED TERMINATION EFFECTS**
- **FULLY TTL AND CMOS COMPATIBLE**

LOGIC SYMBOL



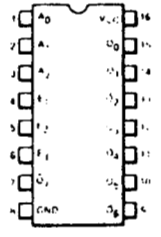
V_{CC} = Pin 16
GND = Pin 8

PIN NAMES

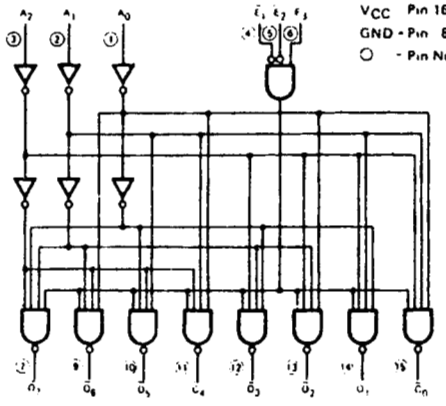
		LOADING (Note a)	
		HIGH	LOW
$A_0 - A_2$	Address Inputs	0.5 U.L.	0.25 U.L.
$\overline{E}_1, \overline{E}_2$	Enable (Active LOW) Inputs	0.5 U.L	0.25 U.L.
E_3	Enable (Active HIGH) Input	0.5 U.L.	0.25 U.L.
$\overline{O}_0 - \overline{O}_7$	Active LOW Outputs (Note b)	10 U.L.	5 (2.5) U.L.

NOTES:
a. 1 TTL Unit Load (U.L.) = 40 µA HIGH/1.6 mA LOW.
b. The Output LOW drive factor is 2.5 U.L. for Military (54) and 5 U.L. for Commercial (74) Temperature Ranges.

CONNECTION DIAGRAM
DIP (TOP VIEW)



LOGIC DIAGRAM

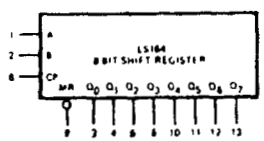


T54LS164/T74LS164
SERIAL-IN PARALLEL-OUT SHIFT REGISTER

DESCRIPTION – The T54LS164/T74LS164 is a high speed 8-Bit Serial-In Parallel-Out Shift Register. Serial data is entered through a 2-Input AND gate synchronous with the LOW to HIGH transition of the clock. The device features an asynchronous Master Reset which clears the register setting all outputs LOW independent of the clock. It utilizes the Schottky diode clamped process to achieve high speeds and is fully compatible with all SGS-ATES TTL products.

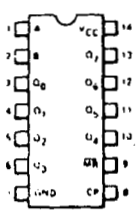
- TYPICAL SHIFT FREQUENCY OF 35 MHz
- ASYNCHRONOUS MASTER RESET
- GATED SERIAL DATA INPUT
- FULLY SYNCHRONOUS DATA TRANSFERS
- INPUT CLAMP DIODES LIMIT HIGH SPEED TERMINATION EFFECTS
- FULLY TTL AND CMOS COMPATIBLE

LOGIC SYMBOL



V_{CC} = Pin 14
GND = Pin 7

CONNECTION DIAGRAM DIP (TOP VIEW)

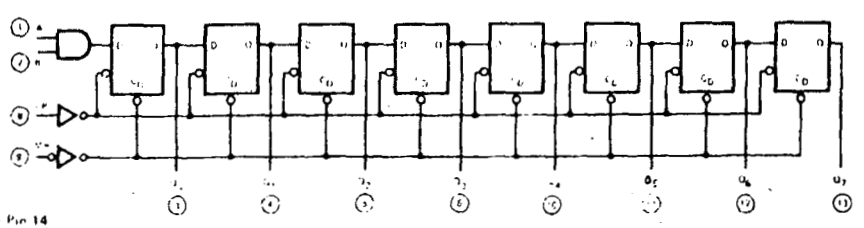


PIN NAMES

		LOADING (Note a)	
		HIGH	LOW
A, B	Data Inputs	0.5 U.L.	0.25 U.L.
CP	Clock (Active HIGH Going Edge) Input	0.5 U.L.	0.25 U.L.
$\overline{MR}$	Master Reset (Active LOW) Input	0.5 U.L.	0.25 U.L.
$Q_0 - Q_7$	Outputs (Note b)	10 U.L.	5(2.5) U.L.

NOTES
a. 1 TTL Unit Load (U.L.) = 40 µA HIGH/1.6 mA LOW.
b. The Output LOW drive factor is 2.5 U.L. for Military (54) and 5 U.L. for Commercial (74) Temperature Ranges.

LOGIC DIAGRAM



V_{CC} = Pin 14
GND = Pin 7
( ) = Pin Numbers

T54LS373/T74LS373
OCTAL TRANSPARENT LATCH WITH 3-STATE OUTPUTS

DESCRIPTION — The T54LS/T74LS373 consists of eight latches with 3-state outputs for bus organized system applications. The flip-flops appear transparent to the data (data changes asynchronously) when Latch Enable (LE) is HIGH. When LE is LOW, the data that meets the set-up times is latched. Data appears on the bus when the Output Enable ($\overline{OE}$) is LOW. When $\overline{OE}$ is HIGH the bus outputs is in the high impedance state.

- **EIGHT LATCHES IN A SINGLE PACKAGE**
- **3-STATE OUTPUTS FOR BUS INTERFACING**
- **HYSTERESIS ON LATCH ENABLE**
- **INPUT CLAMP DIODES LIMIT HIGH SPEED TERMINATION EFFECTS**
- **FULLY CMOS AND TTL COMPATABLE**

ABSOLUTE MAXIMUM RATINGS (above which the useful life may be impaired)

Storage Temperature	−65°C to +150°C
Temperature (Ambient) Under Bias	−55°C to +125°C
V_{CC} Pin Potential to Ground Pin	0.5 V to +7.0 V
*Input Voltage (dc)	0.5 V to +15 V
*Input Current (dc)	30 mA to +5.0 mA
Voltage Applied to Outputs (Output HIGH)	−0.5 V to +10 V
Output Current (dc) (Output LOW)	+50 mA

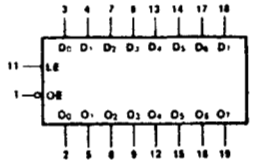
*Either Input Voltage limit or Input Current limit is sufficient to protect the inputs

PIN NAMES

		LOADING (Note a) HIGH	LOW
D_0 - D_7	Data Inputs	0.5 U.L.	0.25 U.L.
LE	Latch Enable (Active HIGH) input	0.5 U.L.	0.25 U.L.
$\overline{OE}$	Output Enable (Active LOW) input	0.5 U.L.	0.25 U.L.
O_0 - O_7	Outputs (Note b)	65 (25) U.L.	15 (7.5) U.L.

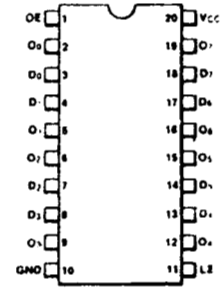
NOTES:
a. 1 TTL Unit Load (U.L.) = 40 μA HIGH/1.6 mA LOW.
b. The output LOW drive factor is 7.5 U.L. for Military (54) and 25 U.L. for Commercial (74) Temperature Ranges. The Output HIGH drive factor is 25 U.L. for Military (54) and 65 U.L. for Commercial (74) Temperature Ranges.

LOGIC SYMBOL

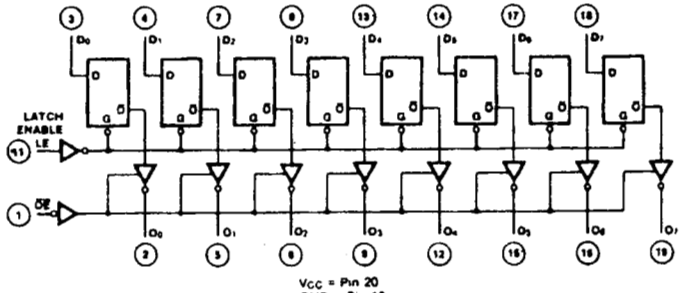


V_{CC} = Pin 20
GND = Pin 10

CONNECTION DIAGRAM DIP (TOP VIEW)



LOGIC DIAGRAM



V_{CC} = Pin 20
GND = Pin 10
○ = Pin Numbers

MOTOROLA
SEMICONDUCTOR
TECHNICAL DATA

μA78S40

UNIVERSAL SWITCHING REGULATOR SUBSYSTEM

The μA78S40 is a monolithic switching regulator subsystem, providing all active functions necessary for a switching regulator system. The device consists of a temperature compensated voltage reference, controlled-duty cycle oscillator with an active current limit circuit, comparator, high-current and high-voltage output switch, capable of 1.5 A and 40 V, pinned-out power diode and an uncommitted operational amplifier, which can be powered up or down independent of the IC supply. The switching output can drive external NPN or PNP transistors when voltages greater than 40 V, or currents in excess of 1.5 A, are required. Some of the features are wide-supply voltage range, low standby current, high efficiency and low drift. The μA78S40 is available in commercial (0°C to +70°C), automotive (−40°C to +85°C), and military (−55°C to +125°C) temperature ranges.

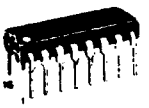
Some of the applications include use in step-up, step-down, and inverting regulators, with extremely good results obtained in battery-operated systems.

- Output Adjustable from 1.25 V to 40 V
- Peak Output Current of 1.5 A Without External Transistor
- 80 dB Line and Load Regulation
- Operation from 2.5 V to 40 V Supply
- Low Standby Current Drain
- High Gain, High Output Current, Uncommitted Op Amp

UNIVERSAL SWITCHING REGULATOR SUBSYSTEM

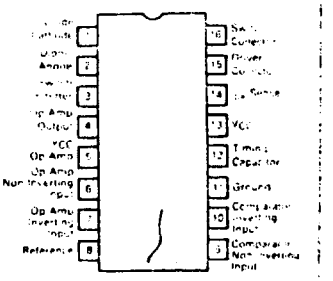
SILICON MONOLITHIC INTEGRATED CIRCUIT

D SUFFIX
CERAMIC PACKAGE
CASE 620



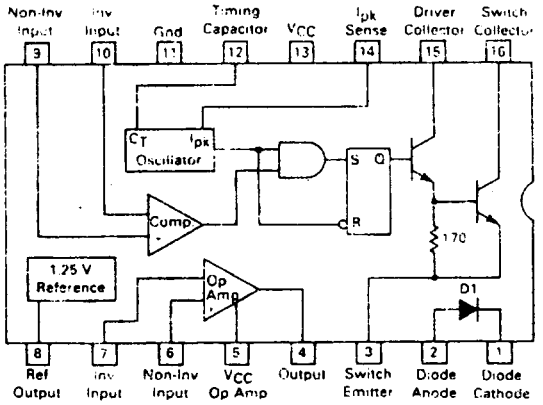
P SUFFIX
PLASTIC PACKAGE
CASE 648

PIN CONNECTIONS



(Top View)

BLOCK DIAGRAM



(Bottom View)

ORDERING INFORMATION

Device	Temperature Range	Package
μA78S40PC	0°C to +70°C	Plastic DIP
μA78S40PV	−40°C to +85°C	Plastic DIP
μA78S40DC	0°C to +70°C	Ceramic DIP
μA78S40DM	−55°C to +125°C	Ceramic DIP

MAXIMUM RATINGS

Rating	Symbol	Value	Unit
Power Supply Voltage	V_{CC}	40	V
Op Amp Power Supply Voltage	V_{CC} (Op Amp)	40	V
Common Mode Input Range (Comparator and Op Amp)	V_{ICR}	-0.3 to V_{CC}	V
Differential Input Voltage (Note 2)	V_{ID}	±30	V
Output Short-Circuit Duration (Op Amp)	—	Continuous	—
Reference Output Current	I_{ref}	10	mA
Voltage from Switch Collectors to Gnd	—	40	V
Voltage from Switch Emitters to Gnd	—	40	V
Voltage from Switch Collectors to Emitter	—	40	V
Voltage from Power Diode to Gnd	—	40	V
Reverse-Power Diode Voltage	V_{DR}	40	V
Current through Power Switch	I_{SW}	1.5	A
Current through Power Diode	I_D	1.5	A
Power Dissipation and Thermal Characteristics			
Plastic Package — T_A = +25°C	P_D	1500	mW
Derate above +25°C (Note 1)	$1/R_{\theta JA}$	14	mW/°C
Ceramic Package — T_A = 25°C	P_D	1000	mW
Derate above +25°C (Note 1)	$1/R_{\theta JA}$	8.0	mW/°C
Storage Temperature Range	T_{stg}	-65 to +150	°C
Operating Temperature Range	T_A		°C
μA78S40M		-55 to +125	
μA78S40V		-40 to +85	
μA78S40C		0 to +70	

Notes:
1. T_{low} = -55°C for μA78S40DM
   -40°C for μA78S40PV
   0°C for μA78S40DC and μA78S40PC
   T_{high} = +125°C for μA78S40DM
   = +85°C for μA78S40PV
   = +70°C for μA78S40DC and μA78S40PC
2. For supply voltages less than 30 V the maximum differential input voltage (Error Amp and Op Amp) is equal to the supply voltage

ELECTRICAL CHARACTERISTICS (V_{CC} = V_{CC} (Op Amp) = 5.0 V, T_A = T_{low} to T_{high} unless otherwise noted.)

Characteristic	Symbol	Min	Typ	Max	Unit
GENERAL					
Supply Voltage	V_{CC}	2.5	—	40	V
Supply Current (Op Amp V_{CC} Disconnected)	I_{CC}				mA
(V_{CC} = 5.0 V)		—	1.8	3.5	
(V_{CC} = 40 V)		—	2.3	5.0	
Supply Current (Op Amp V_{CC} Connected)	I_{CC}				mA
(V_{CC} = 5.0 V)		—	—	4.0	
(V_{CC} = 40 V)		—	—	5.5	
REFERENCE					
Reference Voltage (I_{ref} = 1.0 mA)	V_{ref}	1.180	1.245	1.310	V
Reference Voltage Line Regulation (3.0 V ≤ V_{CC} ≤ 40 V, I_{ref} = 1.0 mA, T_A = 25°C)	Regline	—	0.04	0.2	mV/V
Reference Voltage Load Regulation (1.0 mA ≤ I_{ref} ≤ 10 mA, T_A = 25°C)	Regload	—	0.2	0.5	mV/mA

FIGURE 5 — STEP-DOWN CONVERTER

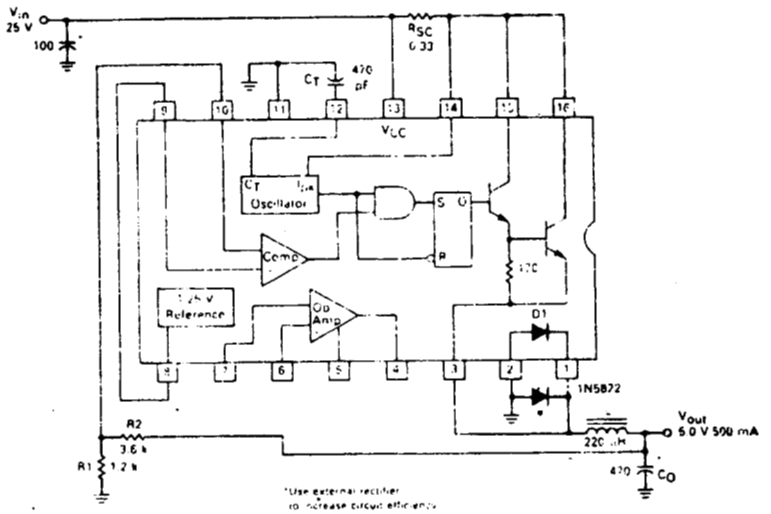


FIGURE 6 — STEP-UP CONVERTER

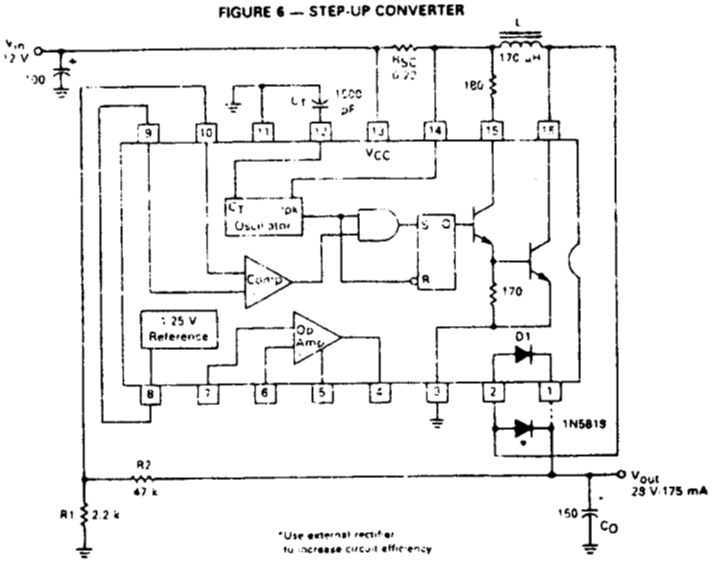


FIGURE 7 — INVERTING CONVERTER

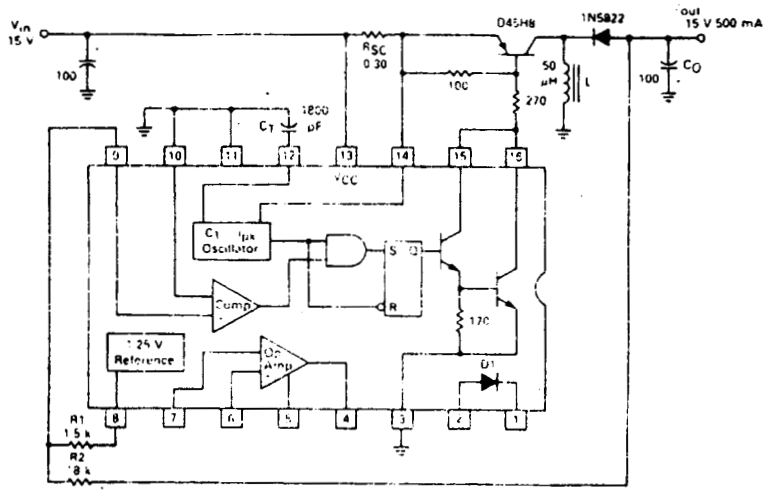


FIGURE 8 — DESIGN FORMULA TABLE

Calculation	Step-Down	Step-Up	Inverting
$\frac{t_{on}}{t_{off}}$	$\frac{V_{out} + V_F}{V_{in(min)} - V_{sat} - V_{out}}$	$\frac{V_{out} + V_F - V_{in(min)}}{V_{in(min)} - V_{sat}}$	$\frac{V_{out} + V_F}{V_{in(min)} - V_{sat}}$
$(t_{on} + t_{off})_{max}$	$\frac{1}{f_{min}}$	$\frac{1}{f_{min}}$	$\frac{1}{f_{min}}$
C_T	$4 \times 10^{-5} t_{on}$	$4 \times 10^{-5} t_{on}$	$4 \times 10^{-5} t_{on}$
$I_{pk(switch)}$	$2 I_{out(max)}$	$2 I_{out(max)} \left(\frac{t_{on} + t_{off}}{t_{off}}\right)$	$2 I_{out(max)} \left(\frac{t_{on} + t_{off}}{t_{off}}\right)$
R_{SC}	$\frac{0.33}{I_{pk(switch)}}$	$\frac{0.33}{I_{pk(switch)}}$	$\frac{0.33}{I_{pk(switch)}}$
$L_{(min)}$	$\left(\frac{V_{in(min)} - V_{sat} - V_{out}}{I_{pk(switch)}}\right) t_{on(max)}$	$\left(\frac{V_{in(min)} - V_{sat}}{I_{pk(switch)}}\right) t_{on(max)}$	$\left(\frac{V_{in(min)} - V_{sat}}{I_{pk(switch)}}\right) t_{on(max)}$
C_O	$\frac{I_{pk(switch)} (t_{on} + t_{off})}{8 V_{ripple(p-p)}}$	$\frac{I_{out} t_{on}}{V_{ripple}}$	$\frac{I_{out} t_{on}}{V_{ripple}}$

V_{sat} = Saturation voltage of the output switch. V_F = Forward voltage drop of the flyback rectifier.

The following power supply characteristics must be chosen:

V_{in} — Nominal input voltage. If this voltage is not constant, then use $V_{in(max)}$ for step-down and $V_{in(min)}$ for step-up and inverting convertor.

V_{out} — Desired output voltage, $V_{out} = 1.25 \left(1 + \frac{R_2}{R_1}\right)$ for step-down and step-up; $V_{out} = \frac{1.25\ R2}{R1}$ for Inverting.

I_{out} — Desired output current.

f_{min} — Minimum desired output switching frequency at the selected values for V_{in} and I_o.

$V_{ripple(p-p)}$ — Desired peak-to-peak output ripple voltage. In practice, the calculated value will need to be increased due to the capacitor's equivalent series resistance and board layout. The ripple voltage should be kept to a low value since it will directly effect the line and load regulation.

See Application Note AN920R2 for further information.

MOTOROLA
SEMICONDUCTOR
TECHNICAL DATA

LM137
LM237
LM337

THREE-TERMINAL ADJUSTABLE OUTPUT NEGATIVE VOLTAGE REGULATORS

The LM137/237/337 are adjustable 3-terminal negative voltage regulators capable of supplying in excess of 1.5 A over an output voltage range of −1.2 V to −37 V. These voltage regulators are exceptionally easy to use and require only two external resistors to set the output voltage. Further, they employ internal current limiting, thermal shutdown and safe area compensation, making them essentially blow-out proof.

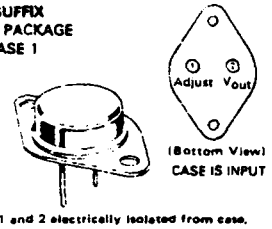
The LM137 series serve a wide variety of applications including local, on-card regulation. This device can also be used to make a programmable output regulator; or, by connecting a fixed resistor between the adjustment and output, the LM137 series can be used as a precision current regulator.

- Output Current in Excess of 1.5 Ampere in K and T Suffix Packages
- Output Current in Excess of 0.5 Ampere In H Suffix Package
- Output Adjustable Between −1.2 V and −37 V
- Internal Thermal Overload Protection
- Internal Short-Circuit-Current Limiting, Constant with Temperature
- Output Transistor Safe-Area Compensation
- Floating Operation for High Voltage Applications
- Standard 3-Lead Transistor Packages
- Eliminates Stocking Many Fixed Voltages

THREE-TERMINAL ADJUSTABLE NEGATIVE VOLTAGE REGULATORS

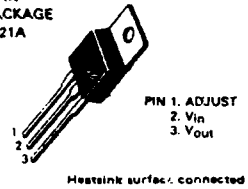
SILICON MONOLITHIC INTEGRATED CIRCUIT

K SUFFIX
METAL PACKAGE
CASE 1

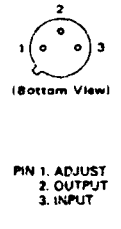


Pins 1 and 2 electrically isolated from case. Case is third electrical connection.

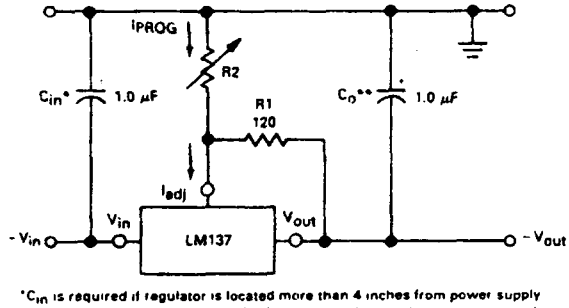
T SUFFIX
PLASTIC PACKAGE
CASE 221A



H SUFFIX
METAL PACKAGE
CASE 79



STANDARD APPLICATION



*C_{in} is required if regulator is located more than 4 inches from power supply filter. A 1 μF solid tantalum or 10 μF aluminum electrolytic is recommended.

**C_O is necessary for stability. A 1 μF solid tantalum or 10 μF aluminum electrolytic is recommended.

$$V_{out} = -1.25\ V\left(1 + \frac{R2}{R1}\right)$$

Automotive temperature range selections are available with special test conditions and additional tests. Contact your local Motorola sales office for information.

ORDERING INFORMATION

Device	Tested Operating Temperature Range	Package
LM137H	T_J = −55°C to +150°C	Metal Can
LM137K		Metal Power
LM237H	T_J = −25°C to +150°C	Metal Can
LM237K		Metal Power
LM337H		Metal Can
LM337K	T_J = 0°C to +125°C	Metal Power
LM337T		Plastic Power
LM337BT*	T_J = −40°C to +125°C	Plastic Power

MOTOROLA
SEMICONDUCTOR
TECHNICAL DATA

LM117
LM217
LM317

THREE-TERMINAL ADJUSTABLE POSITIVE VOLTAGE REGULATORS

SILICON MONOLITHIC INTEGRATED CIRCUIT

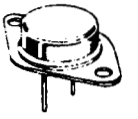
THREE-TERMINAL ADJUSTABLE OUTPUT POSITIVE VOLTAGE REGULATORS

The LM117/217/317 are adjustable 3-terminal positive voltage regulators capable of supplying in excess of 1.5 A over an output voltage range of 1.2 V to 37 V. These voltage regulators are exceptionally easy to use and require only two external resistors to set the output voltage. Further, they employ internal current limiting, thermal shutdown and safe area compensation, making them essentially blow-out proof.

The LM117 series serve a wide variety of applications including local, on card regulation. This device can also be used to make a programmable output regulator, or by connecting a fixed resistor between the adjustment and output, the LM117 series can be used as a precision current regulator.

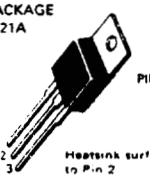
- Output Current in Excess of 1.5 Ampere in K and T Suffix Packages
- Output Current in Excess of 0.5 Ampere in H Suffix Package
- Output Adjustable between 1.2 V and 37 V
- Internal Thermal Overload Protection
- Internal Short-Circuit Current Limiting Constant with Temperature
- Output Transistor Safe-Area Compensation
- Floating Operation for High Voltage Applications
- Standard 3-lead Transistor Packages
- Eliminates Stocking Many Fixed Voltages

K SUFFIX
METAL PACKAGE
CASE 1

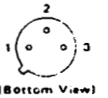


Pins 1 and 2 electrically isolated from case.
Case is third electrical connection

T SUFFIX
PLASTIC PACKAGE
CASE 221A

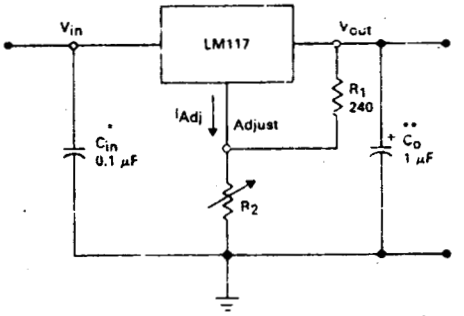


H SUFFIX
METAL PACKAGE
CASE 79



CASE
IS OUTPUT

STANDARD APPLICATION



* C_{in} is required if regulator is located an appreciable distance from power supply filter.

** C_O is not needed for stability, however it does improve transient response.

$$V_{out} = 1.25\ V\ (1 + \frac{R_2}{R_1}) + I_{Adj}\ R_2$$

Since I_{Adj} is controlled to less than 100 µA, the error associated with this term is negligible in most applications.

ORDERING INFORMATION

Device	Tested Operating Temperature Range	Package
LM117H LM117K	T_J = −55°C to +150°C	Metal Can Metal Power
LM217H LM217K	T_J = −25°C to +150°C	Metal Can Metal Power
LM317H LM317K LM317T	T_J = 0°C to +125°C	Metal Can Metal Power Plastic Power
LM317BT#	T_J = −40°C to +125°C	Plastic Power

#Automotive temperature range selections are available with special test conditions and additional tests. Contact your local Motorola sales office for information.

ANEXO 2

LISTADOS DE PROGRAMAS DE PRUEBA

DESCRIPCIÓN.

Debido a la necesidad de estar desarrollando innumerables pruebas para evaluar el funcionamiento de cada etapa del sistema grabador, se precisa una forma ágil y segura para tal efecto; una forma segura pero poco ágil es el uso de memorias EPROM para realizar las pruebas, ya que se requiere tener un grabador y un borrador de EPROM así como de un cierto tiempo para que las memorias se borren lo cual entorpecería el avance del proyecto.

Tomando en cuenta lo anterior, así como la proyección que se da al grabador para poder trabajar mediante el puerto serie de la PC, se realizaron programas con tal fin y adaptando nuestro sistema para aceptar memorias EEPROM (Electrically Erasable and Reprogrammable Only Memories) en lugar de las EPROM para agilizar las pruebas, se desarrollaron los programas "CARGADOR.ASM" Y "CARGAD.PAS".

El programa CARGADOR.ASM se grabo en una EEPROM y este contiene el código necesario para cargar un programa que le llega del puerto serial enviado por el programa CARGAD.PAS a la RAM del grabador y después ejecutarlo. Esto es lo que hizo mas ágiles las pruebas.

En base al hecho de que se debiera tener intercomunicación del sistema con la PC vía puerto serie, se implemento un programa que mostrara dicha intercomunicación, el cual recibe un caracter del teclado de la PC, lo transmite al grabador y este lo regresa para volver a aparecer en la pantalla dos veces. dicho programa lleva por nombre "ECOP.ASM".

Una vez aprobada la intercomunicación, se desarrollan programas para probar y poner a punto la interfaz con el usuario en modo de trabajo independiente (botones y display), surgiendo así el programa "CHKPBS.ASM", permitiendo también el monitoreo de las líneas de datos y direcciones manejando palabras que permitieran examinar el cambio de valor lógico (de 0 a 1 y viceversa) por lo cual se manejaron dos palabras complementarias "AA" y "55", así como los voltajes de programación que se van a manejar, creando el archivo "CHKVTJS.ASM".

Ya teniendo control de todo lo anterior se implementa un programa que además de integrarlo, presenta la interfaz con el usuario, así como los mensajes necesarios para ir llevando al usuario al buen uso de su programador (los cuales se comentan en el manual de usuario) y mostrando un "esqueleto" del programa final, este, enfocado al tipo de EPROM 2764A, surge el programa "GRAB64.ASM" que se ve precedido por todos los anteriores y otros muchos programas de prueba al igual que los arriba mencionados.

Para el modo de trabajo dependiente se desarrollo un programa que permitiera el manejo de las opciones del grabador en este modo, y haciendo uso de algunos de los programas desarrollados en el modo independiente (tales como el cargad.pas, etc) se implementa el programa "ITFC.PAS".

Los códigos fuentes ya compilados y funcionando, se presentan a continuación:

```
0001  0000            ;********* este programa permite cargar un programa en la direccion
0002  0000            ;establecido con el primer .ORG y lo comienza a ejecutar en la direccion
0003  0000            ;propuesta en el segundo .ORG
0004  0000                    #INCLUDE EQUS.TXT
0108+ 0000             .LIST
0005  0000                    .LIST
0006  0000
0007  0000                    .ORG  0000h
0008  0000 21 00               AJMP  INICIO
0009  0002
0010  0100                    .ORG  0100H
0011  0100 74 50      INICIO      MOV   A,#50H      ;PROGRAMA PUERTO SERIE EN MODO 1
0012  0102 F5 98                MOV   SCON,A
0013  0104
0014  0104 74 F4                MOV   A,#0F4H      ;CARGA VALOR DE  BAUDRATE
0015  0106 F5 8D                MOV   TH1,A
0016  0108
0017  0108 74 20                MOV   A,#20H       ;PROGRAMA TIMER 1 EN MODO 2
0018  010A F5 89                MOV   TMOD,A
0019  010C
0020  010C 74 40                MOV   A,#40H       ;ARRANCA TIMER 1
0021  010E F5 88                MOV   TCON,A
0022  0110
0023  0110 31 2C      RECNB       ACALL RECIBE      ;RECIBE EL TOTAL DE BYTES DE CODIGO
0024  0112 FB                   MOV   R3,A
0025  0113 31 2C                ACALL RECIBE       ;R3 CONTIENE BYTE ALTO;
0026  0115 FA                   MOV   R2,A         ;R2 CONTIENE BYTE BAJO;
0027  0116
0028  0116
0029  0116 31 2C      LOOPCAP     ACALL RECIBE      ;RECIBE 2 BYTES DE DIRECCION
0030  0118 F5 83                MOV   DPH,A
0031  011A 31 2C                ACALL RECIBE
0032  011C F5 82                MOV   DPL,A
0033  011E
0034  011E 31 2C                ACALL RECIBE       ;RECIBE  DATO
0035  0120
0036  0120 F0                   MOVX @DPTR,A
0037  0121
0038  0121 1A                   DEC  R2
0039  0122 BA FF F1                CJNE R2,#0FFH,LOOPCAP
0040  0125 1B                   DEC  R3
0041  0126 BB FF ED                CJNE R3,#0FFH,LOOPCAP
0042  0129
0043  0129 02 20 00               LJMP  2000H
0044  012C
0045  012C 30 98 FD    RECIBE      JNB   SCON,0 RECIBE
0046  012F C2 98                 CLR   SCON,0
0047  0131 E5 99                 MOV   A,SBUF
0048  0133 22                   RET
0049  0134
0050  0135                    .ORG $+1          ;OBLIGA AL  ENSAMBLADOR A GENERAR
0051  0135 00 01      RUN         .word INICIO     ;POR SEPARADO UNA LINEA DE CODIGO
0052  0137                    .END              PARA LA DIRECCION DE ARRANQUE.
tasm: Number of errors = 0
```

```
{ESTE PROGRAMA ES EL ENCARGADO DE MANDAR INFORMACION UTILIZANDO EL PUERTO
SERIE 1 DE LA PC; TRABAJA EN CONJUNTO CON EL CARGADOR.ASM}
PROGRAM LEEOBJ;
USES CRT,DOS;
VAR NL,TB,PC,CB,BE,EC,NB,I,J:INTEGER;
   NOMBRE_ARCH,LINEA,TBS,PCS,CBS:STRING[80];
   F,FE:TEXT;
{*****************************************************************}
FUNCTION HEXA (N:INTEGER):STRING;
VAR  LINE :STRING[16];
BEGIN
 LINE:='0123456789ABCDEF';
 HEXA:=LINE[HI(N) DIV 16+1]+LINE[HI(N) MOD 16+1]+
 LINE[LO(N) DIV 16+1]+LINE[LO(N) MOD 16+1];
END;
{********************* ESTE PROCEDIMIENTO MANDA LA INFORMACION CONTENIDA EN EL
REGISTRO 'AL' POR EL PUERTO SERIE HACIENDO USO DE LA INTERRUPCION 14 DEL DOS
*****************************************************************}
PROCEDURE ESCRIBE (X:BYTE);
var
  Reg : Registers;
begin
  with Reg do
  begin
  DX:=0;
  AH:=$01;
  AL:=X;
  INTR($14,REG);
  end;
end;

{*********** ESTE PROCEDIMIENTO ESTABLECE LA CONFIGURACION DEL PUERTO PARA QUE
TRABAJE A LA VELOCIDAD DE 'BAUD RATE' ADECUADA EN BASE AL CRISTAL QUE MANEJA EL
SISTEMA ************************************************}
PROCEDURE INITPORT;
var
  Reg : Registers;
begin
  with Reg do
  begin
   DX:=0;
   AH:=0;
   AL:=$0A3;
   INTR($14,REG);
  end;
end;

BEGIN  {PROGRAMA PRINCIPAL}
 CLRSCR;
 BE:=0;
 NL:=0;
 INITPORT;
 WRITELN;
 WRITE ('NOMBRE DEL ARCHIVO OBJETO:');
 READLN (NOMBRE_ARCH);
 NOMBRE_ARCH:=NOMBRE_ARCH+'.OBJ';
 WRITELN;
```

```
ASSIGN(F,NOMBRE_ARCH);
RESET(F);
TB:=-3;
NL:=-2;
WHILE NOT EOF(F) DO
BEGIN
 READLN(F,LINEA);
 VAL('$'+COPY(LINEA,2,2),NB,EC);
 TB:=TB+NB;
 INC(NL)
END;
TBS:=HEXA(TB);
ESCRIBE(HI(TB));
ESCRIBE(LO(TB));
RESET(F);
FOR J:=1 TO NL DO
BEGIN
 READLN(F,LINEA);
 VAL('$'+COPY(LINEA,2,2),NB,EC);
 VAL('$'+COPY(LINEA,4,4),PC,I);
 FOR I:=1 TO NB DO
 BEGIN
  CBS:=COPY(LINEA,8+I*2,2);
  VAL('$'+CBS,CB,EC);
  PCS:=HEXA(PC);
  GOTOXY(1,6);
  WRITELN('ESCRIBIENDO : $',CBS,' A $',PCS);
  escribe(HI(PC));
  ESCRIBE(LO(PC));
  ESCRIBE(LO(CB));
  INC(BE);
  INC(PC)
 END;
END;
READLN(F,LINEA);
GOTOXY(1,8);
WRITELN('DIRECCION DE ARRANQUE: $',COPY(LINEA,12,2),COPY(LINEA,10,2));
CLOSE(F);
GOTOXY(1,11);
WRITELN('SE ESCRIBIERON $',HEXA(BE),' BYTES.');
END.
```

```
0001  0000             ;este programa regresa el caracter que se introduce desde el teclado de la
0002  0000             ;PC desde un emulador de terminal configurado para transmitir por el
0003  0000             ;puerto serie 1
0004  0000                     #INCLUDE C:EQUS.TXT
0162+ 0000              .LIST
0005  0000                     .LIST
0006  0000
0007  2000                     .ORG  2000h
0008  2000 74 50      INICIO      MOV   A,#50H      ;PROGRAMA PUERTO SERIE EN MODO 1
0009  2002 F5 98                 MOV   SCON,A
0010  2004
0011  2004 74 F4                MOV   A,#0F4H     ;CARGA VALOR DE BAUDRATE
0012  2006 F5 8D                 MOV   TH1,A
0013  2008
0014  2008 74 20                MOV   A,#20H      ;PROGRAMA TIMER 1 EN MODO 2
0015  200A F5 89                 MOV   TMOD,A
0016  200C
0017  200C 74 40                MOV   A,#40H      ;ARRANCA TIMER 1
0018  200E F5 88                 MOV   TCON,A
0019  2010 C2 98                 CLR   SCON.0
0020  2012
0021  2012 11 24      RECNB       ACALL RECIBE      ;RECIBE BYTE
0022  2014
0023  2014 F5 99                 MOV  SBUF,A
0024  2016 30 99 FD   WTBH        JNB  SCON.1,WTBH      ;TRANSMITE ECO
0025  2019 C2 99                 CLR  SCON.1
0026  201B F5 99                 MOV  SBUF,A
0027  201D 30 99 FD   WTBH1       JNB  SCON.1,WTBH1     ;TRANSMITE ECO
0028  2020 C2 99                 CLR  SCON.1
0029  2022 80 EE                 SJMP RECNB
0030  2024
0031  2024
0032  2024 30 98 FD   RECIBE      JNB   SCON.0,RECIBE
0033  2027 C2 98                 CLR   SCON.0
0034  2029 E5 99                 MOV   A,SBUF
0035  202B 22                   RET
0036  202C
0037  202C                      .ORG $+1          ;OBLIGA AL ENSAMBLADOR A GENERAR
0038  202D 00 20      RUN         .word INICIO      ;POR SEPARADO UNA LINEA DE CODIGO
0039  202F                      .END              ;PARA LA DIRECCION DE ARRANQUE.
tasm: Number of errors = 0
```

```
0001  0000   ;ESTE PROGRAMA PERMITE LLER LOS PBS DE LA INTERFAZ CON EL USUARIO
0002  0000   ;ASI COMO DESPLEGAR DATOS EN EL DISPLAY
0003  0000                    #INCLUDE C:EQUS.TXT
0162+ 0000          .LIST
0005  0000                    .LIST
0007  2000                    .ORG    2000H
0009  2000 90 60 03   INICIO:       MOV    DPTR, #PCTRL2
0010  2003 74 88               MOV    A,#88H
0011  2005 F0                 MOVX   @DPTR,A
0012  2006 90 40 03           MOV    DPTR, #PCTRL1
0013  2009 74 80               MOV    A,#80H
0014  200B F0                 MOVX   @DPTR,A
0017  200C 74 DF      REGRESO:      MOV    A,#XCERO
0018  200E 11 2C               ACALL  IMPRIME
0020  2010 90 60 02   ET6:          MOV    DPTR,#PC2
0021  2013 E0                 MOVX   A,@DPTR
0022  2014 54 C0               ANL    A,#0C0H
0023  2016 B4 80 04                  CJNE   A,#080H,ET5
0024  2019 74 86               MOV    A,#XUNO
0025  201B 11 2C               ACALL  IMPRIME
0026  201D
0027  201D 90 60 02   ET5:          MOV    DPTR,#PC2
0028  2020 E0                 MOVX   A,@DPTR
0029  2021 54 C0               ANL    A,#0C0H
0030  2023 B4 40 EA                  CJNE   A,#040H,ET6
0031  2026 74 BB               MOV    A,#XDOS
0032  2028 11 2C               ACALL  IMPRIME
0033  202A
0034  202A 01 10               AJMP   ET6
0038  202C 90 60 00   IMPRIME:      MOV    DPTR,#PA2
0039  202F 78 08               MOV    R0,#08
0040  2031 F9        ETQ4:         MOV    R1,A
0041  2032 54 80               ANL    A,#80H
0042  2034 03                  RR     A
0043  2035 03                  RR     A
0044  2036 44 10               ORL    A,#10H
0045  2038 F0                 MOVX   @DPTR,A
0046  2039 54 EF               ANL    A,#0EFH
0047  203B F0                 MOVX   @DPTR,A
0048  203C E9                  MOV    A,R1
0049  203D 23                  RL     A
0050  203E D8 F1               DJNZ   R0,ETQ4
0051  2040 22                  RET
0052  2041
0053  2041 F0        DATO:         MOVX   @DPTR,A
0054  2042 22                  RET
0055  2043
0056  2044                ORG $+1        ;OBLIGA AL ENSAMBLADOR A GENERAR
0057  2044 00 20     RUN         .WORD   INICIO     ;POR SEPARADO UNA LINEA DE CODIGO
0058  2046                .END           ;PARA LA DIRECCION DE ARRANQUE.
0059  2046
0060  2046
tasm: Number of errors = 0
```

```
0001  0000          ;****** PROGRAMA QUE MANDA LAS PALABRAS DE CONTROL A LOS
                          PUERTOS DE LAS PPIs
0002  0000           ;PARA MONITOREAR LOS VOLTAJES NECESARIOS PARA PROGRAMACION
DE EPROM
0003  0000                  #INCLUDE C:EQUS.TXT
0162+ 0000          .LIST
0004  0000                  .LIST
0005  0000
0006  2000                  .ORG    2000H
0007  2000
0008  2000 90 60 03  INICIO:      MOV    DPTR, #PCTRL2
0009  2003 74 88             MOV    A,#88H
0010  2005 F0                MOVX   @DPTR,A
0011  2006 90 40 03          MOV    DPTR, #PCTRL1
0012  2009 74 80             MOV    A,#80H
0013  200B F0                MOVX   @DPTR,A
0014  200C
0015  200C
0016  200C 74 DF      REGRESO       MOV    A,#XCERO
0017  200E 11 48             ACALL  IMPRIME
0018  2010
0019  2010 90 60 02  ET6:         MOV    DPTR,#PC2
0020  2013 E0                MOVX   A,@DPTR
0021  2014 54 C0             ANL    A,#0C0H
0022  2016 B4 80 12                CJNE   A,#080H,ET5
0023  2019 74 86             MOV    A,#XUNO
0024  201B 11 48             ACALL  IMPRIME
0025  201D 74 AA             MOV    A,#0AAH
0026  201F 90 40 00                MOV    DPTR,#PA1
0027  2022 11 5D             ACALL  DATO
0028  2024 74 17             MOV    A,#017H
0029  2026 90 40 01          MOV    DPTR,#PB1
0030  2029 11 5D             ACALL  DATO
0031  202B
0032  202B 90 60 02  ET5:         MOV    DPTR,#PC2
0033  202E E0                MOVX   A,@DPTR
0034  202F 54 C0             ANL    A,#0C0H
0035  2031 B4 40 DC                CJNE   A,#040H,ET6
0036  2034 74 BB             MOV    A,#XDOS
0037  2036 11 48             ACALL  IMPRIME
0038  2038 74 55             MOV    A,#055H
0039  203A 90 40 00                MOV    DPTR,#PA1
0040  203D 11 5D             ACALL  DATO
0041  203F 74 00             MOV    A,#00H
0042  2041 90 40 01          MOV    DPTR,#PB1
0043  2044 11 5D             ACALL  DATO
0044  2046
0045  2046 01 10             AJMP   ET6
0046  2048
0047  2048
0048  2048
0049  2048 90 60 00  IMPRIME:       MOV    DPTR,#PA2
0050  204B 78 08             MOV    R0,#08
0051  204D F9        ETQ4:          MOV    R1,A
0052  204E 54 80             ANL    A,#80H
0053  2050 03                RR     A
0054  2051 03                RR     A
0055  2052 44 10             ORL    A,#10H
```

```
0056  2054 F0                MOVX   @DPTR,A
0057  2055 54 EF             ANL    A,#0EFH
0058  2057 F0                MOVX   @DPTR,A
0059  2058 E9                MOV    A,R1
0060  2059 23                RL     A
0061  205A D8 F1             DJNZ   R0,ETQ4
0062  205C 22                RET
0063  205D
0064  205D F0        DATO:    MOVX   @DPTR,A
0065  205E 22                RET
0066  205F
0067  2060                ORG $+1        ;OBLIGA AL ENSAMBLADOR A GENERAR
0068  2060 00 20     RUN      .WORD  INICIO    ;POR SEPARADO UNA LINEA DE CODIGO
0069  2062              END          ;PARA LA DIRECCION DE ARRANQUE.
0070  2062
0071  2062
tasm: Number of errors = 0
```

```
/* EL NOMBRE DE ESTE PROGRAMA ES ITFC.PAS QUE ES EL QUE SE ENCARGA DE
INTERFAZAR LA PC CON EL GRABADOR CUANDO SE ESTA TRABAJANDO EN MODO
DEPENDIENTE*/
PROGRAM GRABA;

 USES CRT,DOS;

 CONST
    V12 = '12.5V';
    V21 = '21.0V';
    V25 = '25.0V';
    DIRES= '8000';
    TOPE =65500;

 var AR_INSTS:STRING;
   LINEA:STRING;
   DIRE:STRING[4];
   NNOMAR,NOMAR:STRING[13];
   VDEF,NV,TPE:STRING[7];
   T,S,DIRS,RES:CHAR;
   OP_CODE,C:BYTE;
   SEG_IND:INTEGER;
   NL,TB,PC,CB,BE,EC,NB,I,J:INTEGER;
   UB,N,E,COL,REN,DIR:INTEGER;
   TBS,PCS,CBS:STRING[80];
   F,FE:TEXT;
{   CONTBUF:ARRAY[1..TOPE] OF WORD;}


{*****************************************************************}
FUNCTION HEXAB (N:BYTE):STRING;
VAR  LINE :STRING[16];
BEGIN
 LINE:='0123456789ABCDEF';
 HEXAB:= LINE[ N DIV 16+1]+LINE[ N MOD 16+1];
END;

{*****************************************************************}
FUNCTION HEXA (N:INTEGER):STRING;
VAR  LINE :STRING[16];
BEGIN
 LINE:='0123456789ABCDEF';
 HEXA:=LINE[HI(N) DIV 16+1]+LINE[HI(N) MOD 16+1]+
 LINE[LO(N) DIV 16+1]+LINE[LO(N) MOD 16+1];
END;

{*****************************************************************}
FUNCTION HEXAW (N:WORD):STRING;
VAR  LINE :STRING[16];
BEGIN
 LINE:='0123456789ABCDEF';
 HEXAW:= HEXAB(HI(N))+HEXAB(LO(N));
END;

{*****************************************************************}
FUNCTION BUSCA(OP_CODE:BYTE):INTEGER;
VAR I:INTEGER;
   OP:STRING[2];
```

```
BEGIN
 I:=0;
 OP:=HEXAB(OP_CODE);
 REPEAT
  INC(I);
 UNTIL (COPY(AR_INSTS[I],1,2) = OP) OR (I>242);
 BUSCA:=I;
END;

{*****************************************************************}
PROCEDURE ESCRIBE (X:BYTE);
var
  Reg : Registers;
begin
  with Reg do
  begin
   DX:=0;
   AH:=$01;
   AL:=X;
   INTR($14,REG);
  end;
end;

{*****************************************************************}
PROCEDURE LEE (VAR X:BYTE);
var
  Reg : Registers;
begin

  with Reg do
  begin
   DX:=0;
   AH:=$02;
   INTR($14,REG);
   X:=AL;
  end;

end;

{*****************************************************************}
PROCEDURE INITPORT;
var
  Reg : Registers;
begin
  with Reg do
  begin
   DX:=0;
   AH:=0;
   AL:=$83;
   INTR($14,REG);
  end;
end;

{********************** PROCEDIMIENTO IMPRIME ***************
X,Y =POSICION     PLC =POSICION DE LETRA CAMBIADA  TT = TAMAÑO DEL TEXTO
BS = INDICADOR    LC = LETRA CAMBIADA  ***********************************
*****************************************************************}
PROCEDURE IMPRIME(X,Y:INTEGER;TEXTO:STRING;TT,PLC,BS:INTEGER;LC:CHAR);
```

```
BEGIN
 TEXTCOLOR(0);
 TEXTBACKGROUND(2);
 GOTOXY(X,Y);
 WRITE(TEXTO);
 GOTOXY(PLC,Y);
 TEXTCOLOR(4);
 WRITE(LC);
 GOTOXY(TT,Y);
 IF BS<>0 THEN WRITELN;
 TEXTCOLOR(7);
 TEXTBACKGROUND(0);
END;

{*************************************************************************}
FUNCTION LEE_BYTE(DIR:WORD):BYTE;
VAR
  OP_COD:BYTE;
BEGIN
 ESCRIBE(3);
 ESCRIBE(HI(DIR));
 ESCRIBE(LO(DIR));
 LEE(OP_COD);
 LEE_BYTE:=OP_COD
END;

{********************  PROCEDIMIENTO  MENUE  ************************
INDIC = INDICA SI YA SE ACTIVO LA OPCION O NO, Y CUAL DE ELLAS  *******
*************************************************************************}
PROCEDURE MENUE(INDIC:INTEGER);
BEGIN
 HIGHVIDEO;
 WINDOW (1,2,80,2);
 TEXTBACKGROUND(0);
 TEXTCOLOR(2);
 IF INDIC=1 THEN
   TEXTBACKGROUND(2)
 ELSE
   TEXTBACKGROUND(0);
 WRITE('A');
 TEXTCOLOR(7);
 WRITE('RCHIVO       ');
 TEXTBACKGROUND(0);
 CLREOL;
 IF INDIC=2 THEN
   TEXTBACKGROUND(2)
 ELSE
   TEXTBACKGROUND(0);
 TEXTCOLOR(2);
 WRITE('E');
 TEXTCOLOR(7);
 WRITE('PROM         ');
 TEXTBACKGROUND(0);
 CLREOL;
 IF INDIC=3 THEN
   TEXTBACKGROUND(2)
 ELSE
   TEXTBACKGROUND(0)
```

```
 TEXTCOLOR(2);
 WRITE('B');
 TEXTCOLOR(7);
 WRITE('UFFER          ');
 TEXTBACKGROUND(0);
 CLREOL;
 IF INDIC=4 THEN
 BEGIN
   TEXTBACKGROUND(2);
   WRITE('A');
 END
 ELSE
 BEGIN
   TEXTBACKGROUND(0);
   WRITE('A');
 END;
 TEXTCOLOR(2);
 WRITE('Y');
 TEXTCOLOR(7);
 WRITE('UDA       ');
 TEXTBACKGROUND(0);
 CLREOL;
END;

{*********************  PROCEDIMIENTO ESTADO  ***************************
MANEJA LA "BARRA DE ESTADO"                  ***************************
*************************************************************************}
PROCEDURE ESTADO;
BEGIN
 WINDOW (1,25,80,25);
 GOTOXY(1,1);
 WRITE('ARCHIVO:');
 CLREOL;
 GOTOXY(10,1);
 WRITE(NOMAR);
 GOTOXY(25,1);
 WRITE('TIPO DE EPROM: ');
 CLREOL;
 GOTOXY(39,1);
 WRITE(TPE);
 GOTOXY(50,1);
 WRITE('VOLTAJE: ');
 CLREOL;
 GOTOXY(60,1);
 WRITE(NV);
 GOTOXY(69,1);
 WRITE('SALIR <-+');
END;

{*********************  FUNCION ERROR  ***************************}
PROCEDURE ERROR ;
BEGIN
      SOUND(1000);
      DELAY (500);
      NOSOUND;
      IND:=5;
    END;
```

```
{********** PROCEDIMIENTO CARGA  **********
MANDA AL BUFFER EL ARCHIVO SELECCIONADO  **********
**********}
PROCEDURE CARGA;
BEGIN
 CLRSCR;
 BE:=0;
 NL:=0;
 INITPORT;
 ASSIGN (F,NOMAR);
 RESET (F);
 TB:=-3;
 NL:=-2;

 WHILE NOT EOF(F) DO
 BEGIN
  READLN(F,LINEA);
  VAL('$'+COPY (LINEA,2,2),NB,EC);
  TB:=TB+NB;
  INC(NL);
 END;

 TBS:=HEXA(TB);
 ESCRIBE (HI(TB));
 ESCRIBE (LO(TB));

 RESET(F);

 FOR J:=1 TO NL DO
 BEGIN
  READLN(F,LINEA);
  VAL('$'+COPY(LINEA,2,2),NB,EC);
  VAL('$'+COPY(LINEA,4,4),PC,I);

  WINDOW (20,8,70,15);
  TEXTCOLOR(0);
  TEXTBACKGROUND(2);
  FOR I:=1 TO NB DO
  BEGIN
   CBS:=COPY(LINEA,8+I*2,2);
   VAL('$'+CBS,CB,EC);
   PCS:=HEXA(PC);
   GOTOXY(1,1);
   WRITELN('ESCRIBIENDO : $',CBS,' A  $',PCS);
   escribe(HI(PC));
   ESCRIBE(LO(PC));
   ESCRIBE(LO(CB));
   INC(BE);
   INC(PC);
  END;
 END;

 VAL('$'+COPY(LINEA,12,2),CB,EC);   {MANDA DPH}
 ESCRIBE(CB);

 CLOSE(F);
 GOTOXY(1,5);
 WRITELN('SE ESCRIBIERON  $' HEXA(BE),' BYTES ');
```

```
END;


{********************** PROCEDIMIENTO ARCHI **********************
MANEJA EL MENU DE ARCHIVO      NOMAR =TIENE EL NOMBRE DEL ARCHIVO
*****************************************************************}
PROCEDURE ARCHI;
BEGIN
 MENUE(1);
 WINDOW(1,3,20,7);
 IMPRIME(1,1,'ABRIR          ',15,1,0,'A');
 IMPRIME(1,2,'GUARDAR COMO   ',15,1,0,'G');
 IMPRIME(1,3,'CARGAR A BUFFER ',15,1,0,'C');
 DIRS:=READKEY;
 CLRSCR;
 WINDOW(2,4,33,5);
 CASE DIRS OF
  'A','a': BEGIN
           IMPRIME(1,1,'NOMBRE DEL ARCHIVO: *.OBJ      ',30,28,1,' ');
           GOTOXY(28,1);
           T:=READKEY;
           TEXTCOLOR(0);
           TEXTBACKGROUND(2);
           GOTOXY(21,1);
           CLREOL;
           READLN (NNOMAR);
           NOMAR:=NNOMAR +'.OBJ';
           ASSIGN (F,NOMAR);
           RESET (F);
           CLOSE(F);
          END;
  'G','g': BEGIN
           IMPRIME(1,1,'GUARDAR COMO: *.OBJ          ',30,28,1,' ');
           GOTOXY(28,1);
           T:=READKEY;
           TEXTCOLOR(0);
           TEXTBACKGROUND(2);
           GOTOXY(15,1);
           CLREOL;
           READLN (NNOMAR);
           NOMAR:=NNOMAR +'.OBJ';
{          REWRITE(NOMAR);}
          END;
  'C','c': BEGIN
           WINDOW(34,14,38,16);
           CLRSCR;
           TEXTCOLOR(0);
           TEXTBACKGROUND(2);
           GOTOXY(2,2);
           WRITE('S/N');
           T:=READKEY;
           TEXTCOLOR(7);
           TEXTBACKGROUND(0);
           CLRSCR;
           IF T='S' THEN
             CARGA;
          END
```

```
   ELSE
     ERR[illegible]R;
  END;
 TEXTC[illegible]LOR(7);
 TEXTB[illegible]CKGROUND(0);
 CLRSC[illegible]
END;

{********** ********** PROCEDIMIENTO VOLTAJES  **********
MANEJ[illegible]LA OPCION DE VOLTAJES DE PROGRAMACION
VDEF =[illegible]RESENTA EL VOLTAJE POR DEFAULT     NV = VOLTAJE DE PROGRAMACION
**********[illegible]**********}
PROCE[illegible]URE VOLTAJES;
BEGIN
  WINDO[illegible]W (25,10,65,14);
  TEXTC[illegible]LOR(0);
  TEXTB[illegible]CKGROUND(2);
  IND:=0;
  GOTOX[illegible](1,1);
  WRITE[illegible]N('VOLTAJE DE PROGRAMACION PROPUESTO: ',VDEF,' ');
  GOTO[illegible](1,2);
  WRITE[illegible]N('                                ');
  GOTO[illegible](1,3);
  WRITE[illegible] NUEVO VOLTAJE: ');
  WRITE[illegible]N('S/N                    ');
  GOTO[illegible](16,3);
  RES:=R[illegible]EADKEY;
  CASE [illegible]ES OF
  'S','s':BE[illegible]GIN
          C[illegible]LREOL;
          G[illegible]OTOXY(1,1);
          C[illegible] REOL;
          I[illegible] PRIME(2,1,'(1)12.5V    ',12,3,0,'1');
          I[illegible]PRIME(15,1,'(2)21.0V    ',12,16,0,'2');
          I[illegible] PRIME(28,1,'(3)25.0V  ',10,29,1,'3');
          G[illegible]OTOXY(16,3);
          R[illegible]=READKEY;
          C[illegible]SE S OF
             [illegible]:NV:=V12;
             [illegible]:NV:=V21;
             [illegible]:NV:=V25;
          E[illegible]ND;            {FIN DEL CASE DE S}
          C[illegible]LRSCR;
        E[illegible]ND;              {FIN DEL BEGIN}
  'N','n':N[illegible]V =VDEF
  ELSE
      ERR[illegible]OR;
  END;              {FIN DEL CASE}
END;

{**********[illegible]**********}
PROCE[illegible]URE CONF;
BEGIN
 WINDO[illegible]W(23,4,43,18);
 TEXTC[illegible]LOR(0);
 TEXTBA[illegible]CKGROUND(2);
 IMPRIM[illegible]E (1,2,' (0) 2716    ',12,3,0,'0');
 IMPRIM[illegible]E (1,3,' (1) 2732    ',12,3,0,'1');
```

```
IMPRIME (1,4,' (2) 2732A  ',12,3,0,'2');
IMPRIME (1,5,' (3) 2764   ',12,3,0,'3');
IMPRIME (1,6,' (4) 2764A  ',12,3,0,'4');
IMPRIME (1,7,' (5) 27128  ',12,3,0,'5');
IMPRIME (1,8,' (6) 27128A ',12,3,0,'6');
IMPRIME (1,9,' (7) 27256  ',12,3,0,'7');
IMPRIME(1,10,' (8) 27512  ',12,3,0,'8');
T:=READKEY;
CLRSCR;
IND:=0
CASE T OF
  '0':BEGIN
        TIPE:='2716';
        VDEF:=V25;
        UB:=2047;
        S:='3';
        ESCRIBE(0);
      END;
  '1':BEGIN
        TIPE:='2732';
        VDEF:=V25;
        UB:=4095;
        S:='3';
        ESCRIBE(1);
      END;
  '2':BEGIN
        TIPE:='2732A';
        VDEF:=V21;
        UB:=4096;
        S:='2';
        ESCRIBE(2);
      END;
  '3':BEGIN
        TIPE:='2764';
        VDEF:=V21;
        UB:=8191;
        S:='2';
        ESCRIBE(3);
      END;
  '4':BEGIN
        TIPE:='2764A';
        VDEF:=V12;
        UB:=8191;
        S:='1';
        ESCRIBE(4);
      END;
  '5':BEGIN
        TIPE:='27128';
        VDEF:=V21;
        UB:=16383;
        S:='2';
        ESCRIBE(5);
      END;
  '6':BEGIN
        TIPE:='27128A';
        VDEF:=V12;
        UB:=16383;
        S:='1';
```

```
        ESCRIBE(6);
      END;
   '7':BEGIN
        TIPE:='27256';
        VDEF:=V12;
        UBI:= 32767;
        S:='1';
        ESCRIBE(7);
      END;
   '8':BEGIN
        TIPE:='27512';
        VDEF:=V12;
        UBI:= 6550;
        S:='1';
        ESCRIBE(8);
      END;
 ELSE
  ERROR;
 END;                    {FIN DEL CASE}
 IF IND=0 THEN
   VOLTAJES;
 CASE S OF
    '1':ESCRIBE(0);
    '2':ESCRIBE(1);
    '3':ESCRIBE(2);
 END;            {FIN DEL CASE DE S}
 TEXTCOLOR(7);
 TEXTBACKGROUND(0);
 CLRSCR;
END;

{*****************************************************************************}
PROCEDURE VERIF;
BEGIN
  GOTOXY(1,4);
  WRITELN('SI SIRVE LA VERIFICACION');
  DELAY(1000);
  CLRSCR;
END;

{*****************************************************************************}
PROCEDURE CARG;
BEGIN
  GOTOXY(1,4);
  WRITELN('SI SIRVE LA CARGA');
  DELAY(1000);
  CLRSCR;
END;

{*****************************************************************************}
PROCEDURE IMPR;
BEGIN
  GOTOXY(1,4);
  WRITELN('SI SIRVE LA IMPRESION');
  DELAY(1000);
  CLRSCR;
END;
```

```
{*****************************************************************}
PROCEDURE EPR;
BEGIN
 MENUE(2);
 WINDOW(18,3,44,18);
 IMPRIME(1,1,'PROGRAMAR      ',16,1,0,'P');
 IMPRIME(1,2,'VERIFICAR COPIA ',16,1,0,'V');
 IMPRIME(1,3,'CARGAR A BUFFER ',16,1,0,'C');
 IMPRIME(1,4,'IMPRIMIR        ',16,1,0,'I');
 DIRS:=READKEY;
 CASE DIRS OF
 'p','P':BEGIN
        CLRSCR;
        CONF;
       END
 'v','V':BEGIN
        CLRSCR;
        VERIF;
       END
 'c','C':BEGIN
        CLRSCR;
        CARG;
       END;
 'i','I':BEGIN
        CLRSCR;
        IMPR;
       END;
   ELSE
   BEGIN
      ERROR;
      CLRSCR;
   END;
  END;              {FIN DEL CASE}
END;

{*****************************************************************
CARG_BUF;
BEGIN
 INITPORT;
 ESCRIBE(1);
 VAL('$'+DIRE,DIR,EC);
 DIRE:=HEXAW(DIR);
 ESCRIBE(HI(DIR));
 ESCRIBE(LO(DIR));

 FOR REN :=0 TO UB DC
 BEGIN
    LEE(C);
    CONTBUF[REN+1]:=(C);}
 {END
END
   }
{*****************************************************************}
PROCEDURE EDT;
BEGIN
 WINDOW(19,4,60,21);
 TEXTCOLOR(0);
 TEXTBACKGROUND(2);
```

```
  GOTOXY(1,1);
  WRITELN ('DIRECCION:                    ');
  GOTOXY (12,1);
  READLN(DIRE);
  LOWVIDEO;
  INITPORT;
  ESCRIBE(1);
  VAL('$'+DIRE,DIR,EC);
  DIRE:=HEXAW(DIR);
  ESCRIBE(HI(DIR));
  ESCRIBE(LO(DIR));

  FOR REN :=0 TO 1 DO
  BEGIN
   LOWVIDEO;
   GOTOXY (1,REN+2);
   WRITE(HEXAW(DIR),' ');
   HIGHVIDEO;

   FOR COL:=0 TO 7 DO
   BEGIN
    LEE(C);
    WRITE(HEXAB(C));
    IF COL<15 THEN WRITE(' ');
   END;
   WRITELN('  .....');
   DIR:=DIR+8;
  END;
  GOTOXY(1,20);
  WRITELN'      SALIR <-+                 ');
  REPEAT
   T:=READKEY;
   CASE T OF
     CHR(13):IND:=10;
   ELSE
     ERROR;
   END
  UNTIL IND=10;
END;

{*****************************************************************************}
PROCEDURE BUFF;
BEGIN
 MENU(3);
 WINDOW(35,3,53,6);
 {CARG_EPR;}
 IMPRIME(1,1,'EDITAR      ',9,1,0,'E');
 IMPRIME(1,2,'IMPRIMIR    ',9,1,0,'I');
 DIRS:=READKEY;
 CLRSCR;
 CASE DIRS OF
   'e','E':BEGIN
           IND:=1;
{          T:='S';
           WHILE ((T='S') OR (T='s')) DO
           BEGIN}
            EDT;
            VAL(DIRE,DIR,EC);
```

```
            DIR :=DIR+80;
            STR(DIR,DIRE);}
            TEXTCOLOR(7);
            TEXTBACKGROUND(0);
            CLRSCR;
          END;
   'i':IMPR;
 ELSE
  ERROR;
 END;
 CLRSCR;
END;

{*****************************************************************}
PROCEDURE ACQ;
BEGIN
 TEXTCOLOR(0);
 TEXTBACKGROUND(2);
 WINDOW(20,10,70,16);
 GOTOXY(1,1);
 WRITE('  GRABADOR DE EPROMS PARA EL PROYECTO TERMINAL 2 ');
 GOTOXY(1,2);
 WRITE('                                                ');
 GOTOXY(1,3);
 WRITE('            OCTUBRE DE 1996                     ');
 GOTOXY(1,4);
 WRITE('                                                ');
 GOTOXY(1,5);
 WRITE('            DERECHOS RESERVADOS                 ');
 DELAY(1500);
END;

{*****************************************************************}
PROCEDURE MAN;
BEGIN
 GOTOXY(1,4);
 WRITELN('SI SIRVE EL MANUAL');
 DELAY(1000);
 CLRSCR;
END;

{*****************************************************************}
PROCEDURE AYUD;
BEGIN
 MENU(4);
 WINDOW(52,3,75,6);
 IMPRIME(1,1,'ACERCA DE..',13,1,0,'A');
 IMPRIME(1,2,'MANUAL    ',13,1,0,'M');
 DIRSE:=READKEY;
 CLRSCR;
 CASE DIRS OF
  'a':ACQ;
  'm':MAN;
 ELSE
  ERROR;
 END;
 TEXTCOLOR(7);
 TEXTBACKGROUND(0);
```

```
 CLRSCR;
END.

{*****************************************************************}
PROCEDURE INITPC;
BEGIN
 ESCRIBE(6);
END;

{*****************************************************************}
PROCEDURE INIT;
BEGIN
 TEXTCOLOR(7);
 TEXTBACKGROUND(0);
 CLRSCR;
 MENU(0);
 ESTADO;
END.

{*****************  PROGRAMA PRINCIPAL  ************************
NOMAR : NOMBRE DEL ARCHIVO      TPE = TIPO DE EPROM  ******************
NV   VOLTAJE DE PROGRAMACION  ***************************************
*****************************************************************}
BEGIN
 NOMAR:='*.OBJ';
 TPE:='2716';
 NV:='12.5V';
 INIT
 REPEAT
  FLUSH(INPUT);
  S:=READKEY;
   CASE S OF
   'A','a':ARCHI;
   'E','e':PR;
   'B','b':BUFF;
   'Y','y','N','n':UD;
   #13:IND:=8;
   ELSE
     ERROR;
  END;
  MENU(0);
  ESTADO;
 UNTIL S=CHR(13);
 ESCRIBE(0);
END.
```

Noviembre 96